User's Manual

32

V850E/IG4-H, V850E/IH4-H

User's Manual: Hardware

RENESAS MCU
V850E/Ix4-H Microcontrollers

V850E/IG4-H:
μPD70F3919
μPD70F3920
μPD70F3921

V850E/IH4-H:
μPD70F3922
μPD70F3923
μPD70F3924

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.3.00 September 2011

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

NOTES FOR CMOS DEVICES

① **VOLTAGE APPLICATION WAVEFORM AT INPUT PIN**

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② **HANDLING OF UNUSED INPUT PINS**

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ **PRECAUTION AGAINST ESD**

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ **STATUS BEFORE INITIALIZATION**

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

⑤ **POWER ON/OFF SEQUENCE**

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ **INPUT OF SIGNAL DURING POWER OFF STATE**

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers

This manual is intended for users who wish to understand the functions of the V850E/IG4-H (μPD70F3919, 70F3920, 70F3921) and V850E/IH4-H (μPD70F3922, 70F3923, 70F3924) and design application systems using the V850E/IG4-H and V850E/IH4-H.

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850E/IG4-H and V850E/IH4-H shown in the **Organization** below.

Organization

This manual is divided into two parts: Hardware (this manual) and Architecture (**V850E1 Architecture User's Manual**).

Hardware

- Pin functions
- CPU function
- On-chip peripheral functions
- Flash memory programming
- Electrical specifications

Architecture

- Data types
- Register set
- Instruction format and instruction set
- Interrupts and exceptions
- Pipeline operation

How to Read This Manual

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

To understand the overall functions of the V850E/IG4-H and V850E/IH4-H
→ Read this manual according to the **CONTENTS**.

To find the details of a register where the name is known
→ See **APPENDIX B REGISTER INDEX**.

Register format
→ The name of the bit whose number is in angle brackets (<>) in the figure of the register format of each register is defined as a reserved word in the device file.

To understand the details of an instruction function
→ Refer to the **V850E1 Architecture User's Manual**.

To know the electrical specifications of the V850E/IG4-H and V850E/IH4-H
→ See **CHAPTER 28 ELECTRICAL SPECIFICATIONS**.

The "yyy bit of the xxx register" is described as the "xxx.yyy bit" in this manual. Note with caution that even if "xxx.yyy" is described as is in a program, however, the compiler/assembler cannot recognize it correctly.

The mark <R> shows major revised points.
The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.

Conventions

Data significance: Higher digits on the left and lower digits on the right
Active low representation: ~~xxx~~ (overscore over pin or signal name)
Memory map address: Higher addresses on the top and lower addresses on the bottom
Note: Footnote for item marked with **Note** in the text
Caution: Information requiring particular attention
Remark: Supplementary information
Numeric representation: Binary ... xxxx or xxxxB
Decimal ... xxxx
Hexadecimal ... xxxxH
Prefix indicating power of 2 (address space, memory capacity):
K (kilo): 2^{10} = 1,024
M (mega): 2^{20} = $1,024^{2}$
G (giga): 2^{30} = $1,024^{3}$
Data type: Word … 32 bits
Halfword … 16 bits
Byte … 8 bits

Related Documents

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents related to V850E/IG4-H and V850E/IH4-H

Document Name	Document No.
V850E1 Architecture User's Manual	U14559E
V850E/IG4-H, V850E/IH4-H Hardware User's Manual	This manual

Documents related to development tools (user's manuals)

Document Name		Document No.
QB-V850MINI On-Chip Debug Emulator		U17638E
QB-MINI2 On-Chip Debug Emulator with Programming Function		U18371E
QB-Programmer Programming GUI	Operation	U18527E
CA850 Ver. 3.20 C Compiler Package	Operation	U18512E
	C Language	U18513E
	Assembly Language	U18514E
	Link Directives	U18515E
PM+ Ver. 6.30 Project Manager		U18416E
ID850QB Ver. 3.40 Integrated Debugger	Operation	U18604E
TW850 Ver. 2.00 Performance Analysis Tuning Tool		U17241E
SM+ System Simulator	Operation	U18601E
	User Open Interface	U18212E
RX850 Ver. 3.20 Real-Time OS	Basics	U13430E
	Installation	U17419E
	Technical	U13431E
	Task Debugger	U17420E
RX850 Pro Ver. 3.21 Real-Time OS	Basics	U18165E
	In-Structure	U18164E
	Task Debugger	U17422E
AZ850 Ver. 3.30 System Performance Analyzer		U17423E
PG-FP4 Flash Memory Programmer		U15260E
PG-FP5 Flash Memory Programmer		U18865E

Remark The in-circuit emulator is a product of Midas lab Inc.
For details, contact Midas lab.

Other Documents

Document Name	Document No.
RENESAS MICROCOMPUTER GENERAL CATALOG	R01CS0001E
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the “Semiconductor Device Mount Manual” website (http://www.renesas.com/prod/package/manual/index.html).

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

CONTENTS

V850E/IG4-H, V850E/IH4-H
RENESAS MCU

R01UH0306EJ0300
Rev.3.00
Sep 30, 2011

CHAPTER 1 Introduction

The V850E/IG4-H and V850E/IH4-H are products of the Renesas Electronics V850 single-chip microcontrollers. This chapter gives an outline of the V850E/IG4-H and V850E/IH4-H.

1.1 Overview

The V850E/IG4-H and V850E/IH4-H are 32-bit single-chip microcontrollers that use the V850E1 CPU core and incorporate ROM/RAM and various peripheral functions such as DMA controller, timer/counter, watchdog timer, serial interfaces, USB function controller, A/D converter, and on-chip debug function.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the V850E/IG4-H and V850E/IH4-H feature instructions such as multiply instructions realized by a hardware multiplier, saturated operation instructions, and bit manipulation instructions, as optimum instructions for digital servo control applications. Moreover, as a real-time control system, the V850E/IG4-H and V850E/IH4-H enable an extremely high cost-performance for applications such as motor inverter control.

Table 1-1 lists the V850E/IG4-H and V850E/IH4-H products.

Table 1-1. V850E/IG4-H, V850E/IH4-H Product List

Function Part Number		Package	ROM		RAM Size	Operating Frequency (MAX.)	Maskable Interrupt		Non-Maskable Interrupt
			Type	Size			External	Internal	
V850E/IG4-H	μPD70F3919	100GC	Flash memory	256 KB	24 KB	100 MHz	22	83	1
	μPD70F3920			384 KB					
	μPD70F3921			480 KB					
V850E/IH4-H	μPD70F3922	128GF		256 KB					
	μPD70F3923			384 KB					
	μPD70F3924			480 KB					

Remark 100GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
128GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

Table 1-2 shows the differences in functions between the V850E/IG4-H and V850E/IH4-H.

Table 1-2. Differences in Functions Between V850E/IG4-H and V850E/IH4-H

Item	Separate bus mode	V850E/IG4-H	V850E/IH4-H
Port function	I/O	51	68
	Input	12	12
	On-chip pull-up resistor	51	68
External bus function	Separate bus mode	None	Provided
Timer AB0, timer AB1	Input pin	TIB00 to TIB03	TIB00 to TIB03 TIB10 to TIB13
	Output pin	TOB00 to TOB03 TOB10	TOB00 to TOB03 TOB10 to TOB13
Motor control function	Output pin for 6-phase PWM output mode	TOB0B1 to TOB0B3 TOB1B3 TOB0T1 to TOB0T3 TOB1T3	TOB0B1 to TOB0B3 TOB1B1 to TOB1B3 TOB0T1 to TOB0T3 TOB1T1 to TOB1T3
	6-phase PWM output mode	TAB0 + TMQOP0 (+TAA0)	TAB0 + TMQOP0 (+TAA0) TAB1 + TMQOP0 (+TAA0)
A/D converter 1	Analog input	3 channels	4 channels
On-chip debug function	Trace function	None	Provided
Power supply		EV_{DD0} to EV_{DD2} EV_{SS0} to EV_{SS2}, EV_{SS4} V_{DD0} to V_{DD2} V_{SS0} to V_{SS2} AV_{DD0} to AV_{DD2} AV_{SS0} to AV_{SS2}	EV_{DD0} to EV_{DD3} EV_{SS0} to EV_{SS4} V_{DD0} to V_{DD2} V_{SS0} to V_{SS2} AV_{DD0} to AV_{DD2} AV_{SS0} to AV_{SS2} FV_{DD}
Package		100-pin plastic LQFP (14 × 14)	128-pin plastic LQFP (14 × 20)

1.2 Features

○ Minimum instruction execution time:
10 ns (at internal 100 MHz operation)

○ General-purpose registers: 32 bits × 32

○ CPU features:
Signed multiplication (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits):
1 to 2 clocks
Saturated operation instructions (with overflow/underflow detection function)
32-bit shift instructions: 1 clock
Bit manipulation instructions
Load/store instructions with long/short format
Signed load instructions

○ Memory space:
256 MB linear address space (shared by the program and data)
Chip select output function: 3 spaces[Note]

Note $\overline{\text{CS2}}$ does not exist as an external signal in the V850E/IG4-H and V850E/IH4-H. $\overline{\text{CS2}}$ is used internally as a chip select signal for the USB function area in these products.

Memory block division function: 2 MB/block

- External bus interface

Bus mode
- V850E/IG4-H: Multiplexed bus mode
- V850E/IH4-H: Multiplexed bus mode/separate bus mode

8-/16-bit data bus sizing function
External bus clock frequency (f_{BUS}) = f_{CLK}/4
Wait function
- Programmable wait function
- External wait function

Idle state function
Address setup wait function

○ Internal memory:
RAM: 24 KB (See **Table 1-1**)
Flash memory: 256/384/480 KB (See **Table 1-1**)

○ On-chip debug function: Supports MINICUBE®, MINICUBE2.

○ Interrupts/exceptions:
Non-maskable interrupts: 1 source (external: none, internal: 1)
Maskable interrupts: 105 sources (external: 22, internal: 83)
Software exceptions: 32 sources
Exception traps: 2 sources

○ DMA controller:
7 channels
Transfer unit: 8 bits/16 bits/32 bits
Maximum transfer count: 4096 (2^{12})

Transfer type:	2-cycle
Transfer modes:	Single/single step
Transfer targets:	On-chip peripheral I/O ↔ Internal RAM
Transfer request:	On-chip peripheral I/O/software

Next address setting function

○ I/O lines:
V850E/IG4-H: Total: 63 (Input ports: 12, I/O ports: 51)
V850E/IH4-H: Total: 80 (Input ports: 12, I/O ports: 68)

○ Timer/counter function:
16-bit interval timer M (TMM): 4 channels
16-bit timer/event counter AA (TAA): 3 channels
16-bit timer/event counter AB (TAB): 2 channels
16-bit timer/event counter T (TMT): 4 channels
Motor control function
Timers used in V850E/IG4-H TAB: 1 channel (TAB0), TAA: 1 channel (TAA0)
Timers used in V850E/IH4-H TAB: 2 channels (TAB0, TAB1), TAA: 2 channels (TAA0, TAA1)
16-bit accuracy 6-phase PWM function with deadtime
V850E/IG4-H: 1 channel
V850E/IG4-H: 2 channels
High-impedance output control function
A/D trigger generation by timer tuning operation function
Arbitrary cycle setting function
Arbitrary deadtime setting function
Watchdog timer: 1 channel

○ Serial interfaces:
Asynchronous serial interface A (UARTA)
Asynchronous serial interface B (UARTB)
Clocked serial interface F (CSIF)
I^{2}C bus interface (I^{2}C)
USB function controller (USBF)

UARTA0/CSIF0:	1 channel
UARTA1/I^{2}C:	1 channel
UARTA2/CSIF1:	1 channel
UARTB/CSIF2:	1 channel
USBF:	1 channel

○ A/D converter:

- 12-bit resolution A/D converters (A/D converters 0 and 1)
  V850E/IG4-H: 4 channels + 3 channels (2 units)
  V850E/IH4-H: 4 channels + 4 channels (2 units)
  The three A/D converter 0 channels and three A/D converter 1 channels are provided with an operational amplifier for input level amplification and a comparator for overvoltage detection.
- 10-bit resolution A/D converter (A/D converter 2): 12 channels (1 unit)

○ Clock generator: 10 to 12.5 MHz resonator connectable (external clock input prohibited)

Multiplication function by PLL clock synthesizer (fixed to multiplication by eight, f_{XX} = 80 to 100 MHz)
CPU clock division function (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$)

○ Power-save function: HALT/IDLE/STOP mode

○ Power-on-clear function

○ Low-voltage detection function

○ Package:
- V850E/IG4-H: 100-pin plastic LQFP (fine pitch) (14 × 14)
- V850E/IH4-H: 128-pin plastic LQFP (fine pitch) (14 × 20)

○ Operating supply voltage: When A/D converters 0 to 2 are operating
$V_{DD0} = V_{DD1} = V_{DD2}$ = 1.35 to 1.65 V
FV_{DD} (V850E/IH4-H only) = 4.0 to 5.5 V
$EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3}$ (V850E/IH4-H only) = $AV_{DD0} = AV_{DD1} = AV_{DD2}$ = 4.0 to 5.5 V
When A/D converters 0 to 2 are not operating
$V_{DD0} = V_{DD1} = V_{DD2}$ = 1.35 to 1.65 V
FV_{DD} (V850E/IH4-H only) = 4.0 to 5.5 V
$EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3}$ (V850E/IH4-H only) = $AV_{DD0} = AV_{DD1} = AV_{DD2}$ = 3.5 to 5.5 V

1.3 Application Fields

- Consumer equipment (such as inverter air conditioners, washing machines, driers, refrigerators, etc.)
- Industrial equipment (such as motor control, general-purpose inverters, etc.)

1.4 Ordering Information

1.4.1 V850E/IG4-H

Part Number	Package	Internal ROM (Flash Memory)
μPD70F3919GC-UEU-AX	100-pin plastic LQFP (fine pitch) (14 × 14)	256 KB
μPD70F3920GC-UEU-AX		384 KB
μPD70F3921GC-UEU-AX		480 KB

Remark The V850E/IG4-H microcontrollers are lead-free products.

1.4.2 V850E/IH4-H

Part Number	Package	Internal ROM (Flash Memory)
μPD70F3922GF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	256 KB
μPD70F3923GF-GAT-AX		384 KB
μPD70F3924GF-GAT-AX		480 KB

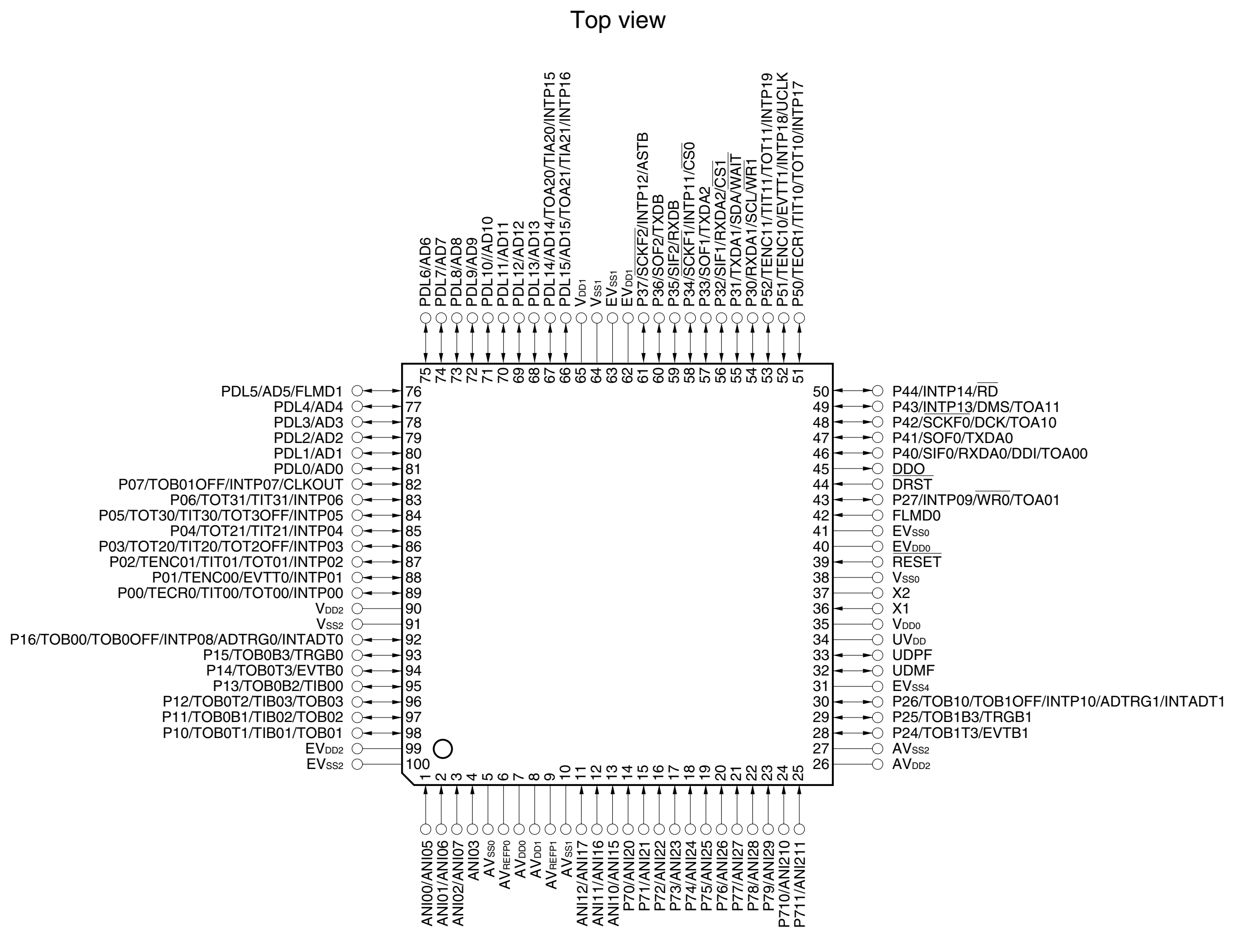
Remark The V850E/IH4-H microcontrollers are lead-free products.

1.5 Pin Configuration

1.5.1 V850E/IG4-H

- **100-pin plastic LQFP (fine pitch) (14 × 14)**

  μPD70F3919GC-UEU-AX
  μPD70F3920GC-UEU-AX
  μPD70F3921GC-UEU-AX



Pin Identification

AD0 to AD15:	Address/data bus
ADTRG0, ADTRG1:	A/D trigger input
ANI00 to ANI07,	
ANI10 to ANI12,	
ANI15 to ANI17,	
ANI20 to ANI211:	Analog input
ASTB:	Address strobe
AV_{DD0} to AV_{DD2}:	Analog power supply
AV_{REFP0}, AV_{REFP1}:	Analog reference voltage
AV_{SS0} to AV_{SS2}:	Analog ground
CLKOUT	Clock output
$\overline{CS0}$, $\overline{CS1}$	Chip select
DCK:	Debug clock
DDI:	Debug data input
DDO:	Debug data output
DMS:	Debug mode select
$\overline{DRST}$:	Debug reset
EV_{DD0} to EV_{DD2}:	Power supply for ports
EV_{SS0} to EV_{SS2}, EV_{SS4}:	Ground for ports
EVTB0, EVTB1,	
EVTT0, EVTT1:	Timer event count input
FLMD0, FLMD1:	Flash programming mode
INTADT0, INTADT1,	
INTP00 to INTP19:	External interrupt input
P00 to P07:	Port 0
P10 to P16:	Port 1
P24 to P27:	Port 2
P30 to P37:	Port 3
P40 to P44:	Port 4
P50 to P52:	Port 5
P70 to P711:	Port 7
PDL0 to PDL15:	Port DL
$\overline{RD}$:	Read strobe
$\overline{RESET}$:	Reset
RXDA0 to RXDA2,	
RXDB:	Receive data
$\overline{SCKF0}$ to $\overline{SCKF2}$:	Serial clock
SCL:	Serial clock
SDA:	Serial data

SIF0 to SIF2:	Serial input
SOF0 to SOF2:	Serial output
TECR0, TECR1:	Timer encoder clear input
TENC00, TENC01,	
TENC10, TENC11:	Timer encoder input
TIA20, TIA21,	
TIB00 to TIB03,	
TIT00, TIT01,	
TIT10, TIT11,	
TIT20, TIT21,	
TIT30, TITI31:	Timer trigger input
TOA00, TOA01,	
TOA10, TOA11,	
TOA20, TOA21,	
TOB00 to TOB03,	
TOB0B1 to TOB0B3,	
TOB0T1 to TOB0T3,	
TOB10, TOB1B3,	
TOB1T3,	
TOT00, TOT01,	
TOT10, TOT11,	
TOT20, TOT21,	
TOT30, TOT31:	Timer output
TOB01OFF,	
TOB0OFF, TOB1OFF,	
TOT2OFF, TOT3OFF:	Timer output off
TRGB0, TRGB1:	Timer trigger input
TXDA0 to TXDA2,	
TXDB:	Transmit data
UCLK:	USB clock
UDMF:	USB data I/O (–) Function
UDPF:	USB data I/O (+) Function
UV_{DD}	Power supply for USB
V_{DD0} to V_{DD2}:	Power supply
V_{SS0} to V_{SS2}:	Ground
$\overline{WAIT}$:	Wait
$\overline{WR0}$, $\overline{WR1}$	Write strobe
X1, X2:	Clock oscillator pins

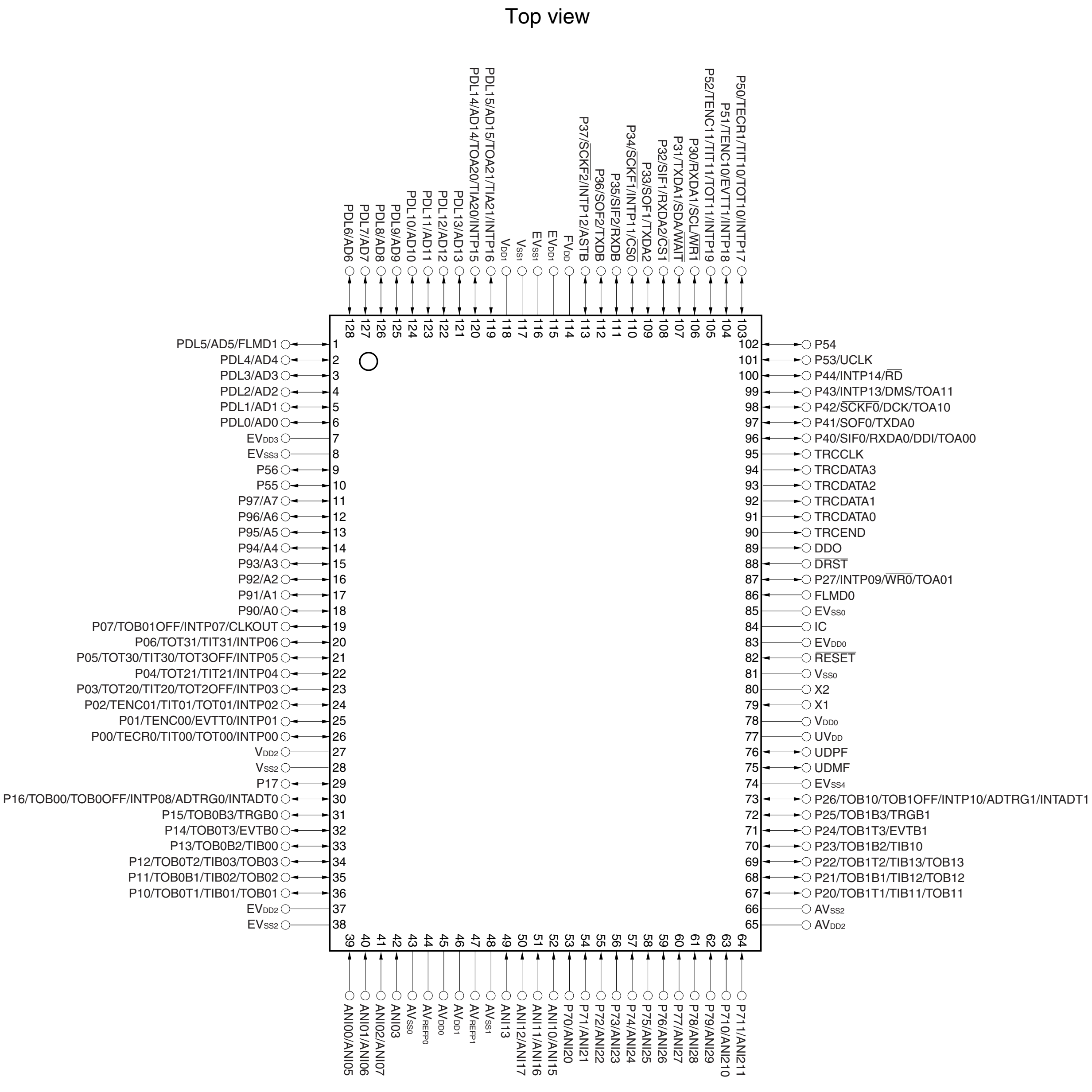
1.5.2 V850E/IH4-H

- **128-pin plastic LQFP (fine pitch) (14 × 20)**

  μPD70F3922GF-GAT-AX

  μPD70F3923GF-GAT-AX

  μPD70F3924GF-GAT-AX



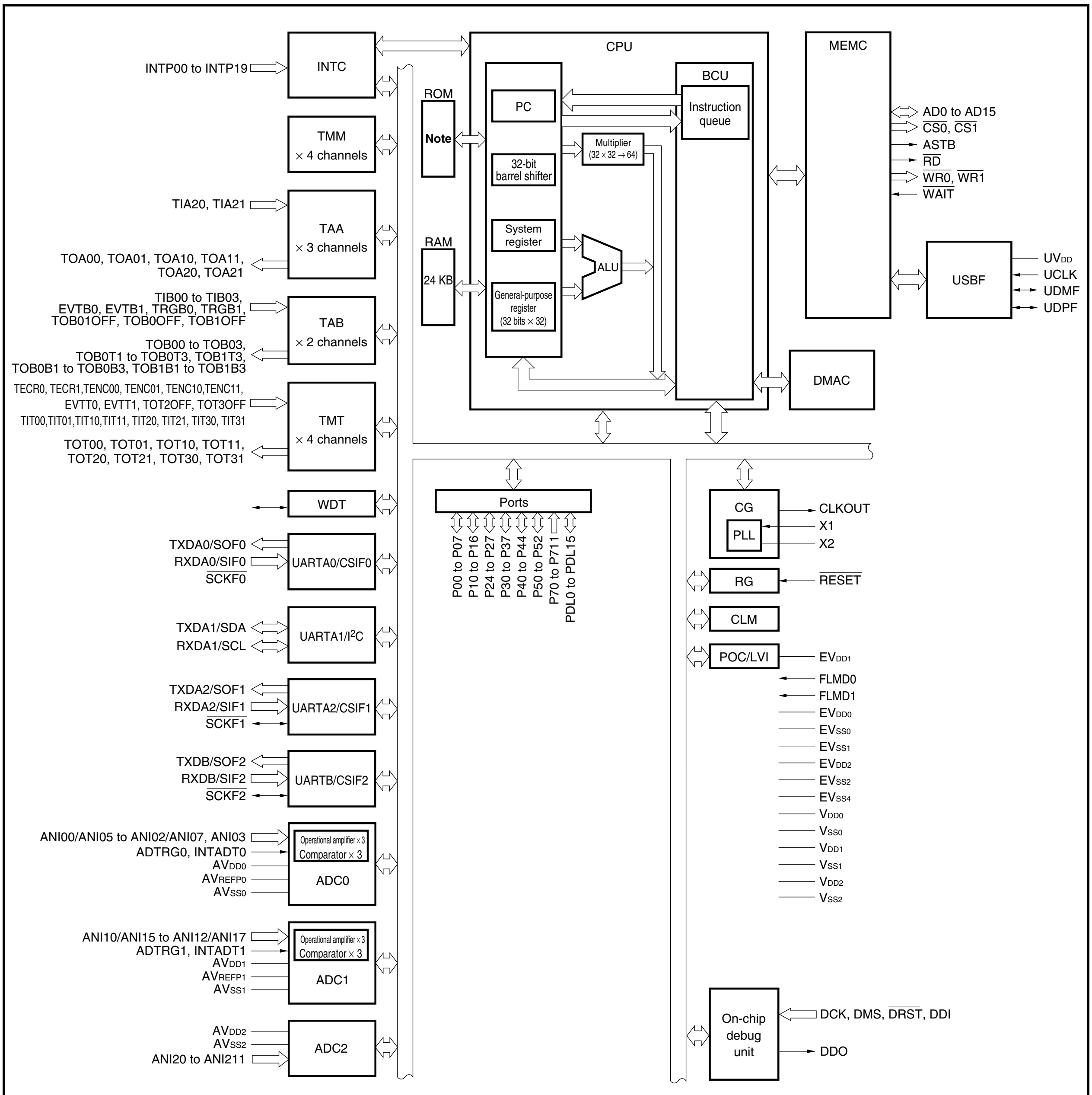
Pin Identification

A0 to A7:	Address bus	SOF0 to SOF2:	Serial output
AD0 to AD15	Address/data bus	TECR0, TECR1:	Timer encoder clear input
ADTRG0, ADTRG1:	A/D trigger input	TENC00, TENC01,	
ANI00 to ANI07,		TENC10, TENC11:	Timer encoder input
ANI10 to ANI17		TIA20, TIA21,	
ANI20 to ANI211:	Analog input	TIB00 to TIB03,	
ASTB:	Address strobe	TIB10 to TIB13,	
AV_{DD0} to AV_{DD2}:	Analog power supply	TIT00, TIT01,	
AV_{REFP0}, AV_{REFP1}:	Analog reference voltage	TIT10, TIT11,	
AV_{SS0} to AV_{SS2}:	Analog ground	TIT20, TIT21,	
CLKOUT:	Clock output	TIT30, TITI31:	Timer trigger input
$\overline{CS0}$, $\overline{CS1}$:	Chip select	TOA00, TOA01,	
DCK:	Debug clock	TOA10, TOA11,	
DDI:	Debug data Input	TOA20, TOA21,	
DDO:	Debug data output	TOB00 to TOB03,	
DMS:	Debug mode select	TOB0B1 to TOB0B3,	
$\overline{DRST}$:	Debug reset	TOB0T1 to TOB0T3,	
EV_{DD0} to EV_{DD3}:	Power supply for ports	TOB10 to TOB13,	
EV_{SS0} to EV_{SS4}:	Ground for ports	TOB1B1 to TOB1B3,	
EVTB0, EVTB1,		TOB1T1 to TOB1T3,	
EVTT0, EVTT1:	Timer event count input	TOT00, TOT01,	
FLMD0, FLMD1:	Flash programming mode	TOT10, TOT11,	
FV_{DD}	Power supply for flash memory	TOT20, TOT21,	
IC	Internally connected	TOT30, TOT31:	Timer output
INTADT0, INTADT1,		TOB01OFF,	
INTP00 to INTP19:	External interrupt input	TOB0OFF, TOB1OFF,	
P00 to P07:	Port 0	TOT2OFF, TOT3OFF:	Timer output off
P10 to P17:	Port 1	TRCCLK	Trace clock
P20 to P27:	Port 2	TRCDATA0 to	
P30 to P37:	Port 3	TRCDATA3:	Trace data output
P40 to P44:	Port 4	TRCEND:	Trace end status output
P50 to P56:	Port 5	TRGB0, TRGB1:	Timer trigger input
P70 to P711:	Port 7	TXDA0 to TXDA2,	
P90 to P97	Port 9	TXDB:	Transmit data
PDL0 to PDL15:	Port DL	UCLK:	USB clock
$\overline{RD}$	Read strobe	UDMF:	USB data I/O (–) function
$\overline{RESET}$:	Reset	UDPF:	USB data I/O (+) function
RXDA0 to RXDA2,		UV_{DD}	Power supply for USB
RXDB:	Receive data	V_{DD0} to V_{DD2}:	Power supply
$\overline{SCKF0}$ to $\overline{SCKF2}$:	Serial clock	V_{SS0} to V_{SS2}:	Ground
SCL:	Serial clock	$\overline{WAIT}$:	Wait
SDA:	Serial data	$\overline{WR0}$, $\overline{WR1}$	Write strobe
SIF0 to SIF2:	Serial input	X1, X2:	Clock oscillator pins

1.6 Function Blocks

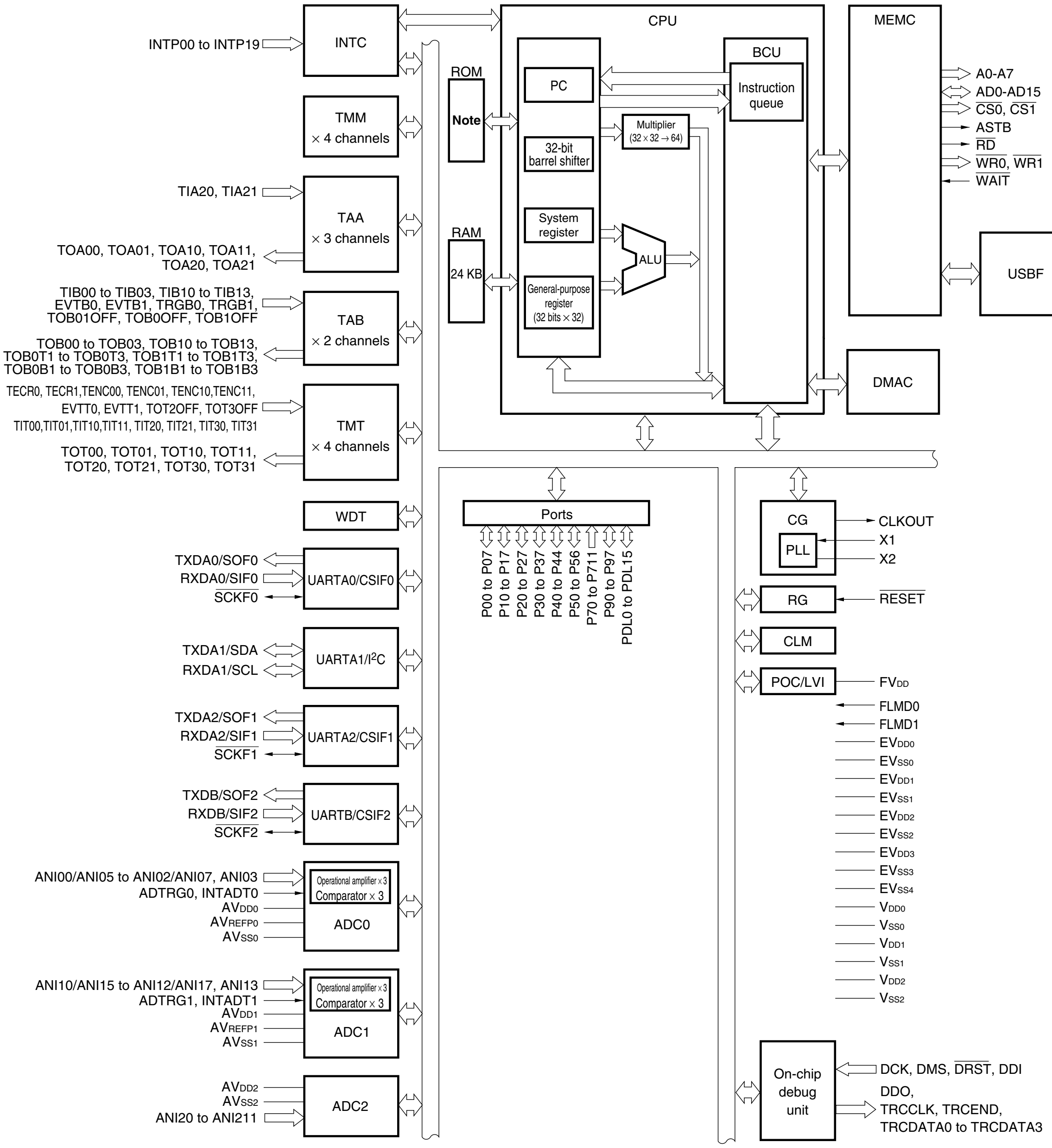
1.6.1 Internal block diagrams

(1) V850E/IG4-H



Note μPD70F3919: 256 KB (flash memory)
μPD70F3920: 384 KB (flash memory)
μPD70F3921: 480 KB (flash memory)

(2) V850E/IH4-H



Note μPD70F3922: 256 KB (flash memory)
μPD70F3923: 384 KB (flash memory)
μPD70F3924: 480 KB (flash memory)

1.6.2 Internal units

(1) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.

Other dedicated on-chip hardware, such as a multiplier (32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate complex processing.

(2) Bus control unit (BCU)

The bus control unit (BCU) starts the required external bus cycles in accordance with the physical address obtained by the CPU. If the CPU does not request the start of a bus cycle when an instruction is fetched from the external memory area, the BCU generates a prefetch address and prefetches an instruction code. The prefetched instruction code is loaded to the CPU's internal instruction queue.

The BCU controls a memory controller (MEMC) via which it accesses the external memory.

(a) Memory controller (MEMC)

The memory controller (MEMC) is used to access the SRAM, external ROM, and external I/O.

(b) DMA controller (DMAC)

This controller controls data transfer between on-chip peripheral I/O and internal RAM in place of the CPU.

The transfer type is two-cycle transfer, and the transfer mode can be selected from single transfer and single-step transfer.

(3) ROM

This is a 480 KB, 384 KB, or 256 KB flash memory that is mapped to addresses 0000000H to 0077FFFH, 0000000H to 005FFFFH, or 0000000H to 003FFFFH, respectively.

During instruction fetch, the ROM can be accessed from the CPU in 1-clock cycles.

(4) RAM

This is a 24 KB RAM that is mapped to addresses FFF9000H to FFFEFFFH.

During instruction fetch or data access, data can be accessed from the CPU in 1-clock cycles.

(5) Interrupt controller (INTC)

This controller handles hardware interrupt requests (INTP00 to INTP19, INTADT0, INTADT1) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed.

(6) Clock generator (CG)

The clock generator includes two basic operation modes: PLL mode (fixed to multiplication by eight) and clock-through mode. It generates four types of clocks (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$), and supplies one of them as the operating clock for the CPU (f_{CPU}).

(7) Timer/counter

The V850E/IG4-H and V850E/IH4-H incorporate four 16-bit interval timer M (TMM) channels, three 16-bit timer/event counter AA (TAA) channels, two 16-bit timer/event counter AB (TAB) channels, and four 16-bit timer/event counter T (TMT) channels, and can measure pulse interval widths or frequency, enable an inverter function for motor control, and output a programmable pulse.

(8) Watchdog timer (WDT)

A watchdog timer is equipped to detect infinite loops, system abnormalities, etc.

It generates a non-maskable interrupt request signal (INTWDT) or internal reset signal (WDTRES) after an overflow occurs.

(9) Serial interfaces

The V850E/IG4-H and V850E/IH4-H incorporate eight serial interface channels: for three asynchronous serial interface A (UARTA) channels, one asynchronous serial interface B (UARTB) channel, three clocked serial interface F (CSIF) channels, and one I^2C bus interface (I^2C) channel. Of these, UARTA0 and CSIF0, UARTA1 and I^2C, UARTA2 and CSIF1, and UARTB and CSIF2 share pins.

For UARTA, data is transferred via the TXDAn and RXDAn pins (n = 0 to 2).

For UARTB, data is transferred via the TXDB and RXDB pins.

For CSIF, data is transferred via the SOFn, SIFn, and $\overline{\text{SCKFn}}$ pins (n = 0 to 2).

For I^2C, data is transferred via the SCL and SDA pins.

USBF transfers data via the UDMF and UDPF pins.

(10) A/D converters (ADC)

Two high-speed, high-resolution 12-bit A/D converters (ADC0, ADC1), which have 4 and 3 channels (V850E/IG4-H) or 4 and 4 channels (V850E/IH4-H) of analog input pins, and one 10-bit A/D converter (ADC2), which has 12 analog input pins, are provided.

ADC0 and ADC1 include three operational amplifiers and three comparators so that these A/D converters can amplify an analog input voltage and detect overvoltage input.

(11) On-chip debug function

An on-chip debug function supporting MINICUBE and MINICUBE2 can be used, so that a simple, inexpensive debug environment can be organized.

(12) Ports

As shown below, the following ports have general-purpose port functions and control pin functions.

Port	I/O	Alternate Function
Port 0	8-bit I/O	Timer/counter I/O, external interrupt input, external bus interface control signal output
Port 1	7-bit I/O (V850E/IG4-H) 8-bit I/O (V850E/IH4-H)	Timer/counter I/O, external trigger input of A/D converter 0, external interrupt input
Port 2	4-bit I/O (V850E/IG4-H) 8-bit I/O (V850E/IH4-H)	Timer/counter I/O, external trigger input of A/D converter 1, external interrupt input, external bus interface control signal output
Port 3	8-bit I/O	Serial interface I/O, external interrupt input, external bus interface control signal I/O.
Port 4	5-bit I/O	Serial interface I/O, timer/counter output, debug input, external interrupt input, external bus interface control signal output
Port 5	3-bit I/O (V850E/IG4-H) 7-bit I/O (V850E/IH4-H)	Timer/counter I/O, external interrupt input, serial interface input
Port 7	12-bit input	A/D converter 2 input
Port 9[Note]	8-bit I/O (V850E/IH4-H)	External bus interface control signal output
Port DL	16-bit I/O	Timer/counter I/O, external interrupt input, flash memory programming mode input signal, external bus interface control signal I/O

Note V850E/IH4-H only

CHAPTER 2 PIN FUNCTIONS

The functions of the pins in the V850E/IG4-H and V850E/IH4-H are listed below. These pins can be divided into port functions and non-port functions according to their function.

2.1 List of Pin Functions

There are three power supplies for the I/O buffer of a pin: AV_{DD2}, EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3} (V850E/IH4-H only), and UV_{DD}. The relationship between each power supply and the pins is shown below.

Table 2-1. I/O Buffer Power Supplies for Each Pin

(a) V850E/IG4-H

Power Supply	Corresponding Pins
AV_{DD2}	P70 to P711
EV_{DD0}, EV_{DD1}, EV_{DD2}	P00 to P07, P10 to P16, P24 to P27, P30 to P37, P40 to P44, P50 to P52, PDL0 to PDL15, $\overline{\text{RESET}}$, DCK, DDI, DDO, DMS, $\overline{\text{DRST}}$
UV_{DD}	UDMF, UDPF

(b) V850E/IH4-H

Power Supply	Corresponding Pins
AV_{DD2}	P70 to P711
EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P44, P50 to P56, P90 to P97, PDL0 to PDL15, $\overline{\text{RESET}}$, DCK, DDI, DDO, DMS, $\overline{\text{DRST}}$, TRCCLK, TRCDATA0 to TRCDATA3, TRCEND
UV_{DD}	UDMF, UDPF

(1) Port functions

(1/4)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
P00	89	26	I/O	Port 0 (See **4.3.1.**) 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOT21 and TOT31 pins go into a high-impedance state).	TECR0/TIT00/TOT00/INTP00
P01	88	25			TENC00/EVTT0/INTP01
P02	87	24			TENC01/TIT01/TOT01/INTP02
P03	86	23			TOT20/TIT20/TOT2OFF/INTP03
P04	85	22			TOT21/TIT21/INTP04
P05	84	21			TOT30/TIT30/TOT3OFF/INTP05
P06	83	20			TOT31/TIT31/INTP06
P07	82	19			TOB01OFF/INTP07/CLKOUT
P10	98	36	I/O	Port 1 (See **4.3.2.**) V850E/IG4-H: 7-bit I/O port V850E/IH4-H: 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOB0B1 to TOB0B3 and TOB0T1 to TOB0T3 pins (output pins of the alternate function) go into a high-impedance state).	TOB0T1/TIB01/TOB01
P11	97	35			TOB0B1/TIB02/TOB02
P12	96	34			TOB0T2/TIB03/TOB03
P13	95	33			TOB0B2/TIB00
P14	94	32			TOB0T3/EVTB0
P15	93	31			TOB0B3/TRGB0
P16	92	30			TOB00/TOB0OFF/INTP08/ADTRG0/INTADT0
P17[Note]	–	29			–
P20[Note]	–	67	I/O	Port 2 (See **4.3.3.**) V850E/IG4-H: 4-bit I/O port V850E/IH4-H: 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOB1B1 (V850E/IH4-H only), TOB1B2 (V850E/IH4-H only), TOB1B3, TOB1T1 (V850E/IH4-H only), TOB1T2 (V850E/IH4-H only), and TOB1T3 pins (output pins of the alternate function) go into a high-impedance state).	TOB1T1[Note]/TIB11[Note]/TOB11[Note]
P21[Note]	–	68			TOB1B1[Note]/TIB12[Note]/TOB12[Note]
P22[Note]	–	69			TOB1T2[Note]/TIB13[Note]/TOB13[Note]
P23[Note]	–	70			TOB1B2[Note]/TIB10[Note]
P24	28	71			TOB1T3/EVTB1
P25	29	72			TOB1B3/TRGB1
P26	30	73			TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1
P27	43	87			INTP09/$\overline{\text{WR0}}$/TOA01

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(2/4)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
P30	54	106	I/O	Port 3 (See **4.3.4.**) 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function (including the $\overline{\text{SCKF1}}$ and $\overline{\text{SCKF2}}$ pins in the slave mode)). If the SCL or SDA pin is selected when the alternate function is to be used, N-ch open-drain output can be specified.	RXDA1/SCL/$\overline{\text{WR1}}$
P31	55	107			TXDA1/SDA/$\overline{\text{WAIT}}$
P32	56	108			SIF1/RXDA2/$\overline{\text{CS1}}$
P33	57	109			SOF1/TXDA2
P34	58	110			$\overline{\text{SCKF1}}$/INTP11/$\overline{\text{CS0}}$
P35	59	111			SIF2/RXDB
P36	60	112			SOF2/TXDB
P37	61	113			$\overline{\text{SCKF2}}$/INTP12/ASTB
P40	46	96	I/O	Port 4 (See **4.3.5.**) 5-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function (including the $\overline{\text{SCKF0}}$ pin in the slave mode)).	SIF0/RXDA0/DDI/TOA00
P41	47	97			SOF0/TXDA0
P42	48	98			$\overline{\text{SCKF0}}$/DCK/TOA10
P43	49	99			INTP13/DMS/TOA11
P44	50	100			INTP14/$\overline{\text{RD}}$
P50	51	103	I/O	Port 5 (See **4.3.6.**) V850E/IG4-H: 3-bit I/O port V850E/IH4-H: 7-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function).	TECR1/TIT10/TOT10/INTP17
P51	52	104			TENC10/EVTT1/INTP18/UCLK$^{\text{Note 2}}$
P52	53	105			TENC11/TIT11/TOT11/INTP19
P53$^{\text{Note 1}}$	–	101			UCLK$^{\text{Note 1}}$
P54$^{\text{Note 1}}$	–	102			–
P55$^{\text{Note 1}}$	–	10			–
P56$^{\text{Note 1}}$	–	9			–

Notes 1. V850E/IH4-H only

2. V850E/IG4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(3/4)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
P70	14	53	Input	Port 7 (See **4.3.7.**) 12-bit input port	ANI20
P71	15	54			ANI21
P72	16	55			ANI22
P73	17	56			ANI23
P74	18	57			ANI24
P75	19	58			ANI25
P76	20	59			ANI26
P77	21	60			ANI27
P78	22	61			ANI28
P79	23	62			ANI29
P710	24	63			ANI210
P711	25	64			ANI211
P90[Note]	–	18	I/O	Port 9 (V850E/IH4-H only) (See **4.3.8.**) 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode).	A0[Note]
P91[Note]	–	17			A1[Note]
P92[Note]	–	16			A2[Note]
P93[Note]	–	15			A3[Note]
P94[Note]	–	14			A4[Note]
P95[Note]	–	13			A5[Note]
P96[Note]	–	12			A6[Note]
P97[Note]	–	11			A7[Note]

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(4/4)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
PDL0	81	6	I/O	Port DL (See **4.3.9.**) 16-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function).	AD0
PDL1	80	5			AD1
PDL2	79	4			AD2
PDL3	78	3			AD3
PDL4	77	2			AD4
PDL5	76	1			AD5/FLMD1
PDL6	75	128			AD6
PDL7	74	127			AD7
PDL8	73	126			AD8
PDL9	72	125			AD9
PDL10	71	124			AD10
PDL11	70	123			AD11
PDL12	69	122			AD12
PDL13	68	121			AD13
PDL14	67	120			AD14/TOA20/TIA20/INTP15
PDL15	66	119			AD15/TOA21/TIA21/INTP16

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(2) Non-port pins

(1/7)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
A0[Note]	–	18	Output	8-bit address bus for external memory	P90[Note]
A1[Note]	–	17			P91[Note]
A2[Note]	–	16			P92[Note]
A3[Note]	–	15			P93[Note]
A4[Note]	–	14			P94[Note]
A5[Note]	–	13			P95[Note]
A6[Note]	–	12			P96[Note]
A7[Note]	–	11			P97[Note]
AD0	81	6	I/O	16-bit address/data bus for external memory	PDL0
AD1	80	5			PDL1
AD2	79	4			PDL2
AD3	78	3			PDL3
AD4	77	2			PDL4
AD5	76	1			PDL5/FLMD1
AD6	75	128			PDL6
AD7	74	127			PDL7
AD8	73	126			PDL8
AD9	72	125			PDL9
AD10	71	124			PDL10
AD11	70	123			PDL11
AD12	69	122			PDL12
AD13	68	121			PDL13
AD14	67	120			PDL14/TOA20/TIA20/INTP15
AD15	66	119			PDL15/TOA21/TIA21/INTP16
ADTRG0	92	30	Input	External trigger input for A/D converter 0	P16/TOB00/TOB0OFF/INTP08/INTADT0
ADTRG1	30	73	Input	External trigger input for A/D converter 1	P26/TOB10/TOB1OFF/INTP10/INTADT1
ANI00	1	39	Input	Analog input for A/D converter 0	ANI05
ANI01	2	40			ANI06
ANI02	3	41			ANI07
ANI03	4	42			–
ANI05	1	39			ANI00
ANI06	2	40			ANI01
ANI07	3	41			ANI02

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(2/7)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
ANI10	13	52	Input	Analog input for A/D converter 1	ANI15
ANI11	12	51			ANI16
ANI12	11	50			ANI17
ANI13[Note]	–	49			–
ANI15	13	52			ANI10
ANI16	12	51			ANI11
ANI17	11	50			ANI12
ANI20	14	53	Input	Analog input for A/D converter 2	P70
ANI21	15	54			P71
ANI22	16	55			P72
ANI23	17	56			P73
ANI24	18	57			P74
ANI25	19	58			P75
ANI26	20	59			P76
ANI27	21	60			P77
ANI28	22	61			P78
ANI29	23	62			P79
ANI210	24	63			P710
ANI211	25	64			P711
ASTB	61	113	Output	Address strobe output for external data bus	P37/$\overline{\text{SCKF2}}$/INTP12
AV_{DD0}	7	45	–	Positive power supply for A/D converter 0	–
AV_{DD1}	8	46	–	Positive power supply for A/D converter 1	–
AV_{DD2}	26	65	–	Positive power supply for A/D converter 2	–
AV_{REFP0}	6	44	–	Reference voltage input for A/D converter 0	–
AV_{REFP1}	9	47	–	Reference voltage input for A/D converter 1	–
AV_{SS0}	5	43	–	Ground potential for A/D converter 0	–
AV_{SS1}	10	48	–	Ground potential for A/D converter 1	–
AV_{SS2}	27	66	–	Ground potential for A/D converter 2	–
CLKOUT	82	19	Output	External bus clock output	P07/TOB01OFF/INTP07
$\overline{\text{CS0}}$	58	110	Output	Chip select output	P34/$\overline{\text{SCKF1}}$/INTP11
$\overline{\text{CS1}}$	56	108			P32/SIF1/RXDA2
DCK	48	98	Input	Debug clock input for on-chip debug emulator	P42/$\overline{\text{SCKF0}}$/TOA10
DDI	46	96	Input	Debug data input for on-chip debug emulator	P40/SIF0/RXDA0/TOA00

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(3/7)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
DDO	45	89	Output	Debug data output for on-chip debug emulator	–
DMS	49	99	Input	Debug mode select for on-chip debug emulator	P43/INTP13/TOA11
$\overline{\text{DRST}}$	44	88	Input	Debug reset input for on-chip debug emulator	–
EV_{DD0}	40	83	–	Positive power supply for external pins	–
EV_{DD1}	62	115			–
EV_{DD2}	99	37			–
EV_{DD3}[Note 1]	–	7			–
EV_{SS0}	41	85	–	Ground potential for external pins	–
EV_{SS1}	63	116			–
EV_{SS2}	100	38			–
EV_{SS3}[Note 1]	–	8			–
EV_{SS4}	31	74			–
EVTB0	94	32	Input	External event count input of TAB0, TAB1	P14/TOB0T3
EVTB1	28	71			P24/TOB1T3
EVTT0	88	25	Input	External event count input of TMT0, TMT1/ external trigger input	P01/TENC00/INTP01
EVTT1	52	104			P51/TENC10/INTP18/UCLK[Note 2]
FLMD0	42	86	Input	Pins for setting flash memory programming mode	–
FLMD1	76	1			PDL5/AD5
FV_{DD}[Note 1]	–	114	–	Positive power supply for flash memory	–
IC[Note 1]	–	84	–	Internally connected pins	–
INTADT0	92	30	Input	External maskable interrupt request input	P16/TOB00/TOB0OFF/INTP08/ADTRG0
INTADT1	30	73			P26/TOB10/TOB1OFF/INTP10/ADTRG1
INTP00	89	26			P00/TECR0/TIT00/TOT00
INTP01	88	25			P01/TENC00/EVTT0
INTP02	87	24			P02/TENC01/TIT01/TOT01
INTP03	86	23			P03/TOT20/TIT20/TOT2OFF
INTP04	85	22			P04/TOT21/TIT21
INTP05	84	21			P05/TOT30/TIT30/TOT3OFF
INTP06	83	20			P06/TOT31/TIT31
INTP07	82	19			P07/TOB01OFF/CLKOUT
INTP08	92	30			P16/TOB00/TOB0OFF/ADTRG0/INTADT0

Notes 1. V850E/IH4-H only

2. V850E/IG4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(4/7)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
INTP09	43	87	Input	External maskable interrupt request input	P27/$\overline{\text{WR0}}$/TOA01
INTP10	30	73			P26/TOB10/TOB1OFF/ADTRG1/INTADT1
INTP11	58	110			P34/$\overline{\text{SCKF1}}$/$\overline{\text{CS0}}$
INTP12	61	113			P37/$\overline{\text{SCKF2}}$/ASTB
INTP13	49	99			P43/DMS/TOA11
INTP14	50	100			P44/$\overline{\text{RD}}$
INTP15	67	120			PDL14/AD14/TOA20/TIA20
INTP16	66	119			PDL15/AD15/TOA21/TIA21
INTP17	51	103			P50/TECR1/TIT10/TOT10
INTP18	52	104			P51/TENC10/EVTT1/UCLK$^{\text{Note}}$
INTP19	53	105			P52/TENC11/TIT11/TOT11
$\overline{\text{RD}}$	50	100	Output	Read strobe output of external data bus	P44/INTP14
$\overline{\text{RESET}}$	39	82	Input	System reset input	–
RXDA0	46	96	Input	Serial receive data input of UARTA0 to UARTA2	P40/SIF0/DDI/TOA00
RXDA1	54	106			P30/SCL/$\overline{\text{WR1}}$
RXDA2	56	108			P32/SIF1/$\overline{\text{CS1}}$
RXDB	59	111	Input	Serial receive data input of UARTB0	P35/SIF2
$\overline{\text{SCKF0}}$	48	98	I/O	Serial clock I/O of CSIF0 to CSIF2	P42/DCK/TOA10
$\overline{\text{SCKF1}}$	58	110			P34/INTP11/$\overline{\text{CS0}}$
$\overline{\text{SCKF2}}$	61	113			P37/INTP12/ASTB
SCL	54	106	I/O	Serial clock I/O	P30/RXDA1/$\overline{\text{WR1}}$
SDA	55	107	I/O	Serial transmit/receive data I/O	P31/TXDA1/$\overline{\text{WAIT}}$
SIF0	46	96	Input	Serial receive data input of CSIF0 to CSIF2	P40/RXDA0/DDI/TOA00
SIF1	56	108			P32/RXDA2/$\overline{\text{CS1}}$
SIF2	59	111			P35/RXDB
SOF0	47	97	Output	Serial transmit data output of CSIF0 to CSIF2	P41/TXDA0
SOF1	57	109			P33/TXDA2
SOF2	60	112			P36/TXDB
TECR0	89	26	Input	Encoder clear input of TMT0, TMT1	P00/TIT00/TOT00/INTP00
TECR1	51	103			P50/TIT10/TOT10/INTP17
TENC00	88	25	Input	Encoder input of TMT0, TMT1	P01/EVTT0/INTP01
TENC01	87	24			P02/TIT01/TOT01/INTP02
TENC10	52	104			P51/EVTT1/INTP18/UCLK$^{\text{Note}}$
TENC11	53	105			P52/TIT11/TOT11/INTP19

Note V850E/IG4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(5/7)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
TIA20	67	120	Input	External event count input/external trigger input/ capture trigger input of TAA2	PDL14/AD14/TOA20/INTP15
TIA21	66	119	Input	Capture trigger input of TAA2	PDL15/AD15/TOA21/INTP16
TIB00	95	33	Input	Capture trigger input of TAB0, TAB1	P13/TOB0B2
TIB01	98	36			P10/TOB0T1/TOB01
TIB02	97	35			P11/TOB0B1/TOB02
TIB03	96	34			P12/TOB0T2/TOB03
TIB10[Note]	–	70			P23[Note]/TOB1B2[Note]
TIB11[Note]	–	67			P20[Note]/TOB1T1[Note]/TOB11[Note]
TIB12[Note]	–	68			P21[Note]/TOB1B1[Note]/TOB12[Note]
TIB13[Note]	–	69			P22[Note]/TOB1T2[Note]/TOB13[Note]
TIT00	89	26	Input	Capture trigger input of TMT0, TMT1	P00/TECR0/TOT00/INTP00
TIT01	87	24			P02/TENC01/TOT01/INTP02
TIT10	51	103			P50/TECR1/TOT10/INTP17
TIT11	53	105			P52/TENC11/TOT11/INTP19
TIT20	86	23	Input	External event count input/external trigger input/ capture trigger input of TMT2	P03/TOT20/TOT2OFF/INTP03
TIT21	85	22	Input	Capture trigger input of TMT2	P04/TOT21/INTP04
TIT30	84	21	Input	External event count input/external trigger input/ capture trigger input of TMT3	P05/TOT30/TOT3OFF/INTP05
TIT31	83	20	Input	Capture trigger input of TMT3	P06/TOT31/INTP06
TOA00	46	96	Output	Timer output of TAA0 to TAA2	P40/SIF0/RXDA0/DDI
TOA01	43	87			P27/INTP09/$\overline{\text{WR0}}$
TOA10	48	98			P42/$\overline{\text{SCKF0}}$/DCK
TOA11	49	99			P43/INTP13/DMS
TOA20	67	120			PDL14/AD14/TIA20/INTP15
TOA21	66	119			PDL15/AD15/TIA21/INTP16
TOB00	92	30	Output	Timer output of TAB0	P16/TOB0OFF/INTP08/ADTRG0/INTADT0
TOB01	98	36			P10/TOB0T1/TIB01
TOB01OFF	82	19	Input	High-impedance output control signal input	P07/INTP07/CLKOUT
TOB02	97	35	Output	Timer output of TAB0	P11/TOB0B1/TIB02
TOB03	96	34			P12/TOB0T2/TIB03
TOB0B1	97	35	Output	Pulse signal output for 6-phase PWM low arm of TAB0	P11/TIB02/TOB02
TOB0B2	95	33			P13/TIB00
TOB0B3	93	31			P15/TRGB0
TOB0OFF	92	30	Input	High-impedance output control signal input	P16/TOB00/INTP08/ADTRG0/INTADT0

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(6/7)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
TOB0T1	98	36	Output	Pulse signal output for 6-phase PWM high arm of TAB0	P10/TIB01/TOB01
TOB0T2	96	34			P12/TIB03/TOB03
TOB0T3	94	32			P14/EVTB0
TOB10	30	73	Output	Timer output of TAB1	P26/TOB1OFF/INTP10/ADTRG1/INTADT1
TOB11[Note]	–	67			P20[Note]/TOB1T1[Note]/TIB11[Note]
TOB12[Note]	–	68			P21[Note]/TOB1B1[Note]/TIB12[Note]
TOB13[Note]	–	69			P22[Note]/TOB1T2[Note]/TIB13[Note]
TOB1B1[Note]	–	68	Output	Pulse signal output for 6-phase PWM low arm of TAB1	P21[Note]/TIB12[Note]/TOB12[Note]
TOB1B2[Note]	–	70			P23[Note]/TIB10[Note]
TOB1B3	29	72			P25/TRGB1
TOB1OFF	30	73	Input	High-impedance output control signal input	P26/TOB10/INTP10/ADTRG1/INTADT1
TOB1T1[Note]	–	67	Output	Pulse signal output for 6-phase PWM high arm of TAB1	P20[Note]/TIB11[Note]/TOB11[Note]
TOB1T2[Note]	–	69			P22[Note]/TIB13[Note]/TOB13[Note]
TOB1T3	28	71			P24/EVTB1
TOT00	89	26	Output	Timer output of TMT0 to TMT2	P00/TECR0/TIT00/INTP00
TOT01	87	24			P02/TENC01/TIT01/INTP02
TOT10	51	103			P50/TECR1/TIT10/INTP17
TOT11	53	105			P52/TENC11/TIT11/INTP19
TOT20	86	23			P03/TIT20/TOT2OFF/INTP03
TOT21	85	22			P04/TIT21/INTP04
TOT2OFF	86	23	Input	High-impedance output control signal input	P03/TOT20/TIT20/INTP03
TOT30	84	21	Output	Timer output of TMT3	P05/TIT30/TOT3OFF/INTP05
TOT31	83	20			P06/TIT31/INTP06
TOT3OFF	84	21	Input	High-impedance output control signal input	P05/TOT30/TIT30/INTP05
TRCCLK[Note]	–	95	Output	Trace clock output	–
TRCDATA0[Note]	–	91	Output	Trace data output (D0 to D3)	–
TRCDATA1[Note]	–	92			–
TRCDATA2[Note]	–	93			–
TRCDATA3[Note]	–	94			–
TRCEND[Note]	–	90	Output	Trace end status output	–
TRGB0	93	31	Input	External trigger input of TAB0, TAB1	P15/TOB0B3
TRGB1	29	72			P25/TOB1B3
TXDA0	47	97	Output	Serial transmit data output of UARTA0 to UARTA2	P41/SOF0
TXDA1	55	107			P31/SDA/$\overline{\text{WAIT}}$
TXDA2	57	109			P33/SOF1

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(7/7)

Function Name	Pin No.		I/O	Function	Alternate Function
	IG4-H	IH4-H			
	GC	GF			
TXDB	60	112	Output	Serial transmit data output of UARTB0	P36/SOF2
UCLK	52	–	Input	USB clock signal input	P51/TENC10/EVTT1/INTP18
	–	101			P53[Note]
UDMF	32	75	I/O	USB data I/O (–) function	–
UDPF	33	76		USB data I/O (+) function	–
UV_{DD}	34	77	–	3.3 V positive power supply for USB	–
V_{DD0}	35	78	–	Positive power supply for internal units	–
V_{DD1}	65	118			–
V_{DD2}	90	27			–
V_{SS0}	38	81	–	Ground potential for internal units	–
V_{SS1}	64	117			–
V_{SS2}	91	28			–
$\overline{\text{WAIT}}$	55	107	Input	External wait request input	P31/TXDA1/SDA
$\overline{\text{WR0}}$	43	87	Output	Write strobe output of external data bus	P27INTP09/TOA01
$\overline{\text{WR1}}$	54	106			P30/RXDA1/SCL
X1	36	79	Input	Pins for connecting resonator for system clock	–
X2	37	80	–		–

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

2.2 Pin I/O Circuits and Recommended Connection of Unused Pins

It is recommended to use 1 to 10 kΩ resistors when connecting to AV_{SS2}, EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3} (V850E/IH4-H only), EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), EV_{SS4} or V_{SS0}, V_{SS1}, or V_{SS2} by way of resistors.

(1/4)

Function	Alternate-Function Name	Pin No.		I/O Circuit Type	Recommended Connection of Unused Pins
		IG4-H	IH4-H		
		GC	GF		
P00	TECR0/TIT00/TOT00/INTP00	89	26	5-AH	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}[Note] or EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3}[Note], EV_{SS4} by way of resistors. Output: Leave open.
P01	TENC00/EVTT0/INTP01	88	25		
P02	TENC01/TIT01/TOT01/INTP02	87	24		
P03	TOT20/TIT20/TOT2OFF/INTP03	86	23		
P04	TOT21/TIT21/INTP04	85	22		
P05	TOT30/TIT30/TOT3OFF/INTP05	84	21		
P06	TOT31/TIT31/INTP06	83	20		
P07	TOB01OFF/INTP07/CLKOUT	82	19		
P10	TOB0T1/TIB01/TOB01	98	36		
P11	TOB0B1/TIB02/TOB02	97	35		
P12	TOB0T2/TIB03/TOB03	96	34		
P13	TOB0B2/TIB00	95	33		
P14	TOB0T3/EVTB0	94	32		
P15	TOB0B3/TRGB0	93	31		
P16	TOB00/TOB0OFF/INTP08/ADTRG0/INTADT0	92	30		
P17[Note]	–	–	29	5-AG	
P20[Note]	TOB1T1[Note]/TIB11[Note]/TOB11[Note]	–	67	5-AH	
P21[Note]	TOB1B1[Note]/TIB12[Note]/TOB12[Note]	–	68		
P22[Note]	TOB1T2[Note]/TIB13[Note]/TOB13[Note]	–	69		
P23[Note]	TOB1B2[Note]/TIB10[Note]	–	70		
P24	TOB1T3/EVTB1	28	71		
P25	TOB1B3/TRGB1	29	72		
P26	TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1	30	73		
P27	INTP09/$\overline{\text{WR0}}$/TOA01	43	87		
P30	RXDA1/SCL/$\overline{\text{WR1}}$	54	106		
P31	TXDA1/SDA/$\overline{\text{WAIT}}$	55	107		
P32	SIF1/RXDA2/$\overline{\text{CS1}}$	56	108		
P33	SOF1/TXDA2	57	109	5-AG	

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(2/4)

Function	Alternate-Function Name	Pin No.		I/O Circuit Type	Recommended Connection of Unused Pins
		IG4-H	IH4-H		
		GC	GF		
P34	$\overline{\text{SCKF1}}$/INTP11/$\overline{\text{CS0}}$	58	110	5-AH	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}[Note 1] or EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3}[Note 1], EV_{SS4} by way of resistors. Output: Leave open.
P35	SIF2/RXDB	59	111		
P36	SOF2/TXDB	60	112	5-AG	
P37	$\overline{\text{SCKF2}}$/INTP12/ASTB	61	113	5-AH	
P40	SIF0/RXDA0/DDI/TOA00	46	96		
P41	SOF0/TXDA0	47	97	5-AG	
P42	$\overline{\text{SCKF0}}$/DCK/TOA10	48	98	5-AH	
P43	INTP13/DMS/TOA11	49	99		
P44	INTP14/$\overline{\text{RD}}$	50	100		
P50	TECR1/TIT10/TOT10/INTP17	51	103		
P51	TENC10/EVTT1/INTP18/UCLK[Note 2]	52	104		
P52	TENC11/TIT11/TOT11/INTP19	53	105		
P53[Note 1]	UCLK[Note 1]	–	101		
P54[Note 1]	–	–	102	5-AG	
P55[Note 1]	–	–	10		
P56[Note 1]	–	–	9		
P70	ANI20	14	53	11-G	Independently connect to AV_{SS2} by way of resistors.
P71	ANI21	15	54		
P72	ANI22	16	55		
P73	ANI23	17	56		
P74	ANI24	18	57		
P75	ANI25	19	58		
P76	ANI26	20	59		
P77	ANI27	21	60		
P78	ANI28	22	61		
P79	ANI29	23	62		
P710	ANI210	24	63		
P711	ANI211	25	64		
P90[Note 1]	A0[Note 1]	–	18	5-AG	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}[Note 1] or EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3}[Note 1], EV_{SS4} by way of resistors. Output: Leave open.
P91[Note 1]	A1[Note 1]	–	17		
P92[Note 1]	A2[Note 1]	–	16		
P93[Note 1]	A3[Note 1]	–	15		
P94[Note 1]	A4[Note 1]	–	14		

Notes 1. V850E/IH4-H only

2. V850E/IG4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(3/4)

Function	Alternate-Function Name	Pin No.		I/O Circuit Type	Recommended Connection of Unused Pins
		IG4-H	IH4-H		
		GC	GF		
P95[Note]	A5[Note]	–	13	5-AG	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}[Note] or EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3}[Note], EV_{SS4} by way of resistors. Output: Leave open.
P96[Note]	A6[Note]	–	12		
P97[Note]	A7[Note]	–	11		
PDL0	AD0	81	6		
PDL1	AD1	80	5		
PDL2	AD2	79	4		
PDL3	AD3	78	3		
PDL4	AD4	77	2		
PDL5	AD5/FLMD1	76	1		
PDL6	AD6	75	128		
PDL7	AD7	74	127		
PDL8	AD8	73	126		
PDL9	AD9	72	125		
PDL10	AD10	71	124		
PDL11	AD11	70	123		
PDL12	AD12	69	122		
PDL13	AD13	68	121		
PDL14	AD14/TOA20/TIA20/INTP15	67	120		
PDL15	AD15/TOA21/TIA21/INTP16	66	119		
ANI00	ANI05	1	39	7-C	Connect to AV_{SS0} or AV_{SS1}.
ANI01	ANI06	2	40		
ANI02	ANI07	3	41		
ANI03	–	4	42		
ANI10	ANI15	13	52		
ANI11	ANI16	12	51		
ANI12	ANI17	11	50		
ANI13[Note]	–	–	49		
DDO	–	45	89	3-C	Leave open (output when $\overline{\text{DRST}}$ is high-level).
$\overline{\text{DRST}}$	–	44	88	2-M	Leave open (because a pull-down resistor is on chip).

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(4/4)

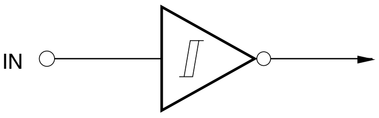
Function	Alternate-Function Name	Pin No.		I/O Circuit Type	Recommended Connection of Unused Pins
		IG4-H	IH4-H		
		GC	GF		
FLMD0	–	42	86	2	–
IC[Note]	–	–	84	2	Always connect to V_{SS0}, V_{SS1}, or V_{SS2}.
$\overline{\text{RESET}}$	–	39	82	2	Pull this pin up when the power-on-clear circuit (POC) is used.
TRCCLK[Note]	–	–	95	3-C	Leave open.
TRCDATA0[Note]	–	–	91		
TRCDATA1[Note]	–	–	92		
TRCDATA2[Note]	–	–	93		
TRCDATA3[Note]	–	–	94		
TRCEND[Note]	–	–	90		
UDMF	–	32	75	–	Always connect to V_{SS0}, V_{SS1}, or V_{SS2} (even in standby mode).
UDPF	–	33	76	–	
UV_{DD}	–	34	77	–	Always connect to V_{DD0}, V_{DD1}, or V_{DD2} (even in standby mode).

Note V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

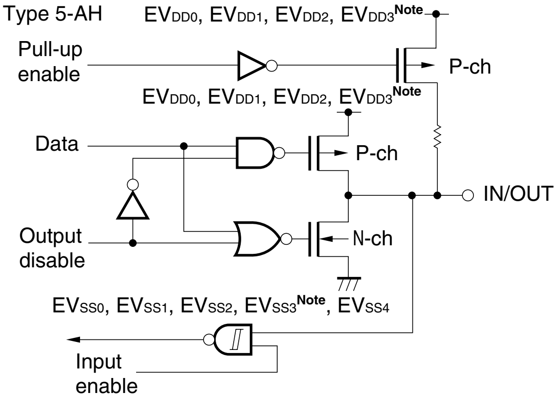
2.3 Pin I/O Circuits

Type 2

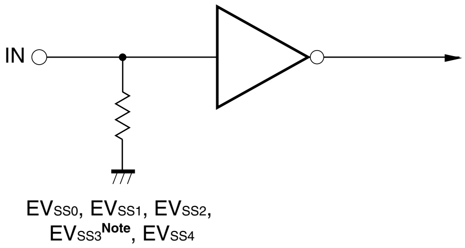


Schmitt-triggered input with hysteresis characteristics

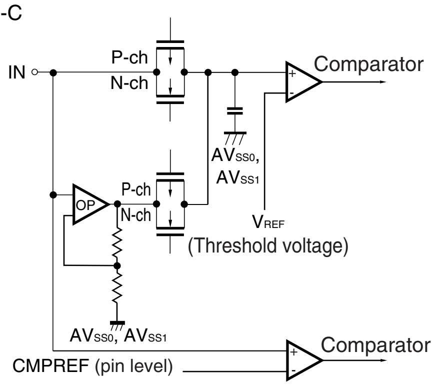
Type 5-AH


EVDD0, EVDD1, EVDD2, EVDD3Note
Pull-up enable
P-ch
EVDD0, EVDD1, EVDD2, EVDD3Note
Data
P-ch
IN/OUT
Output disable
N-ch
EVSS0, EVSS1, EVSS2, EVSS3Note, EVSS4
Input enable

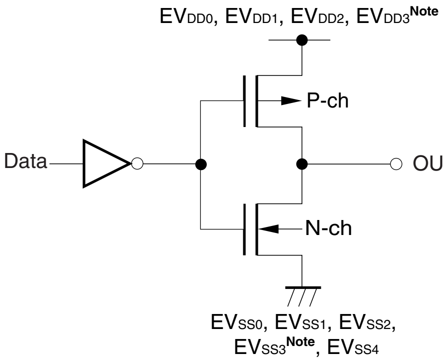
Type 2-M



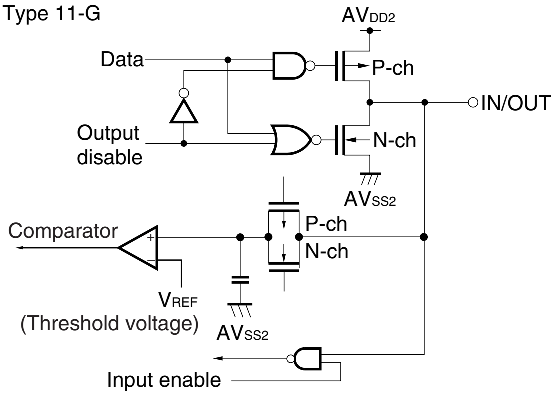
Type 7-C



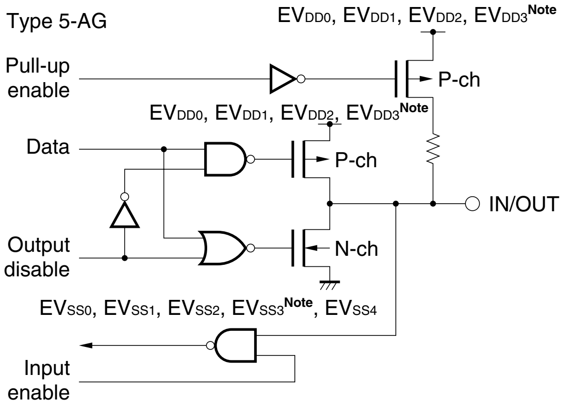
Type 3-C



Type 11-G



Type 5-AG


EVDD0, EVDD1, EVDD2, EVDD3Note
Pull-up enable
P-ch
EVDD0, EVDD1, EVDD2, EVDD3Note
Data
P-ch
IN/OUT
Output disable
N-ch
EVSS0, EVSS1, EVSS2, EVSS3Note, EVSS4
Input enable

Note V850E/IH4-H only

CHAPTER 3 CPU FUNCTION

The CPU of the V850E/IG4-H and V850E/IH4-H is based on RISC architecture and executes almost all the instructions in one clock cycle using 5-stage pipeline control.

3.1 Features

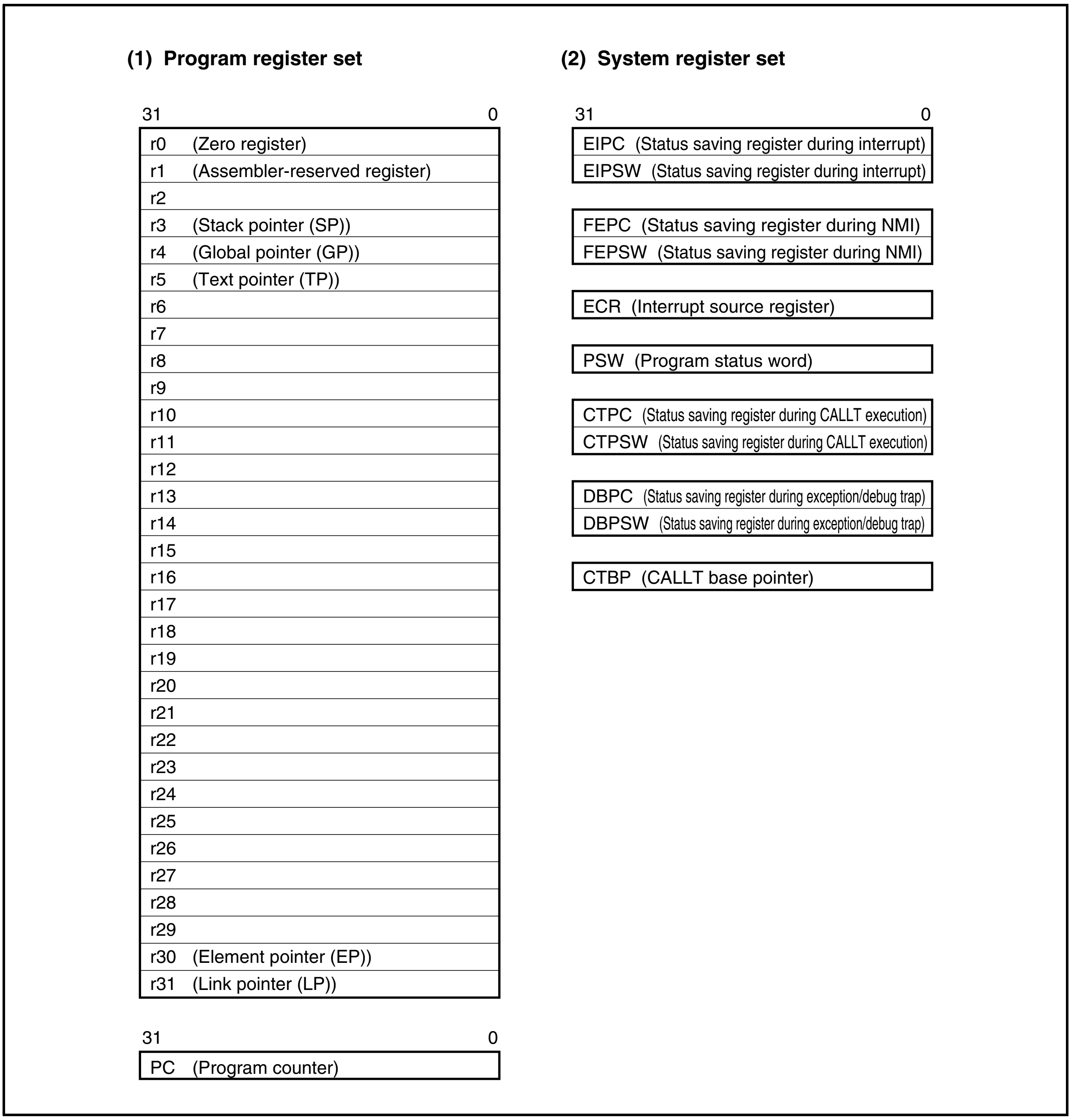
- ○ Minimum instruction execution time: 10 ns (at 100 MHz internal operation)
- ○ Thirty-two 32-bit general-purpose registers
- ○ Internal 32-bit architecture
- ○ Five-stage pipeline control
- ○ Multiply/divide instructions
- ○ Saturated operation instructions
- ○ One-clock 32-bit shift instruction
- ○ Load/store instruction with long/short instruction format
- ○ Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The registers of the V850E/IG4-H and V850E/IH4-H can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers have a 32-bit width.

For details, refer to **V850E1 Architecture User’s Manual**.

Figure 3-1. CPU Register Set



(1) Program register set

31	0
r0	(Zero register)
r1	(Assembler-reserved register)
r2	
r3	(Stack pointer (SP))
r4	(Global pointer (GP))
r5	(Text pointer (TP))
r6	
r7	
r8	
r9	
r10	
r11	
r12	
r13	
r14	
r15	
r16	
r17	
r18	
r19	
r20	
r21	
r22	
r23	
r24	
r25	
r26	
r27	
r28	
r29	
r30	(Element pointer (EP))
r31	(Link pointer (LP))

31	0
PC	(Program counter)

(2) System register set

31	0
EIPC	(Status saving register during interrupt)
EIPSW	(Status saving register during interrupt)
FEPC	(Status saving register during NMI)
FEPSW	(Status saving register during NMI)
ECR	(Interrupt source register)
PSW	(Program status word)
CTPC	(Status saving register during CALLT execution)
CTPSW	(Status saving register during CALLT execution)
DBPC	(Status saving register during exception/debug trap)
DBPSW	(Status saving register during exception/debug trap)
CTBP	(CALLT base pointer)

3.2.1 Program register set

The program register set includes general-purpose registers and a program counter.

(1) General-purpose registers (r0 to r31)

Thirty-two general-purpose registers, r0 to r31, are available. Any of these registers can be used as a data variable or address variable.

However, r0 and r30 are implicitly used by instructions, and care must be exercised when using these registers. r0 is a register that always holds 0, and is used for operations using 0 and offset 0 addressing. r30 is used, by means of the SLD and SST instructions, as a base pointer for when memory is accessed. Also, r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost. The contents must be restored to the registers after the registers have been used. r2 may be used by the real-time OS. If the real-time OS does not use r2, it can be used as a variable register.

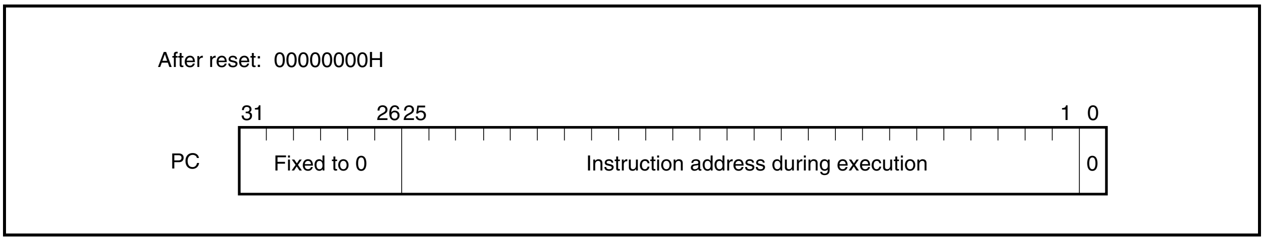
Table 3-1. General-Purpose Registers

Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate data
r2	Address/data variable register (when r2 is not used by the real-time OS)	
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area (where program code is located)
r6 to r29	Address/data variable registers	
r30	Element pointer	Base pointer when memory is accessed
r31	Link pointer	Used by compiler when calling function

(2) Program counter (PC)

This register holds the instruction address during program execution. The lower 26 bits of this register are valid, and bits 31 to 26 are fixed to 0. If a carry occurs from bit 25 to 26, it is ignored.

Bit 0 is fixed to 0, and branching to an odd address cannot be performed.



3.2.2 System register set

System registers control the status of the CPU and hold interrupt information.

To read/write these system registers, specify a system register number indicated below using the system register load/store instruction (LDSR or STSR instruction).

Table 3-2. System Register Numbers

System Register No.	System Register Name	Operand Specification	
		LDSR Instruction	STSR Instruction
0	Interrupt status saving register (EIPC)[Note 1]	√	√
1	Interrupt status saving register (EIPSW)[Note 1]	√	√
2	NMI status saving register (FEPC)	√	√
3	NMI status saving register (FEPSW)	√	√
4	Interrupt source register (ECR)	×	√
5	Program status word (PSW)	√	√
6 to 15	Reserved for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×
16	CALLT execution status saving register (CTPC)	√	√
17	CALLT execution status saving register (CTPSW)	√	√
18	Exception/debug trap status saving register (DBPC)	√[Note 2]	√[Note 2]
19	Exception/debug trap status saving register (DBPSW)	√[Note 2]	√[Note 2]
20	CALLT base pointer (CTBP)	√	√
21 to 31	Reserved for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×

Notes 1. Because this register has only one set, to enable multiple interrupts, it is necessary to save this register by program.

2. These registers can be read/written only in the period between DBTRAP instruction or illegal opcode execution and DBRET instruction execution.

Caution Even if bit 0 of EIPC, FEPC, or CTPC is set to 1 by the LDSR instruction, bit 0 will be ignored when the program is returned by the RETI instruction after interrupt servicing (because bit 0 of the PC is fixed to 0). When setting the value of EIPC, FEPC, and CTPC, use an even value (bit 0 = 0).

Remark √: Access allowed

×: Access prohibited

(1) Interrupt status saving registers (EIPC, EIPSW)

There are two interrupt status saving registers, EIPC and EIPSW.

Upon occurrence of a software exception or a maskable interrupt, the contents of the program counter (PC) are saved to EIPC and the contents of the program status word (PSW) are saved to EIPSW (upon occurrence of a non-maskable interrupt (NMI), the contents are saved to the NMI status saving registers (FEPC, FEPSW)).

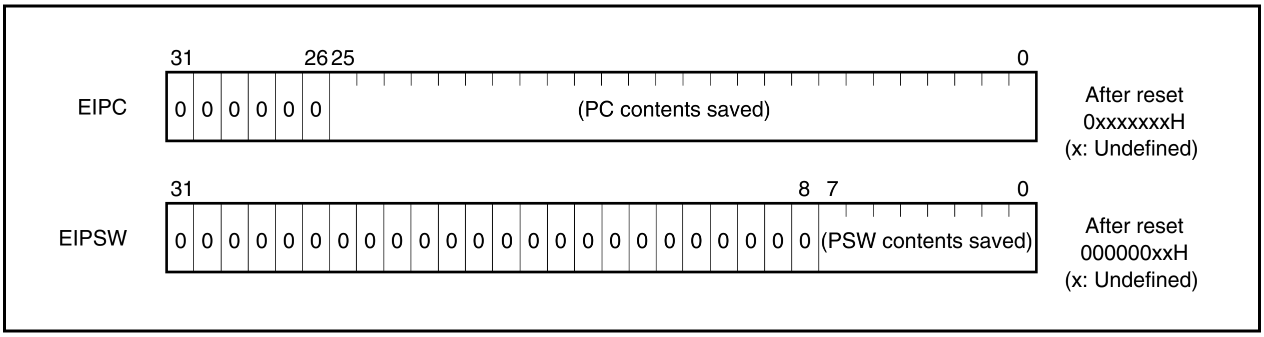
The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for some instructions (see **21.9 Periods in Which CPU Does Not Acknowledge Interrupts**).

The current PSW contents are saved to EIPSW.

Since there is only one set of interrupt status saving registers, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.

Bits 31 to 26 of EIPC and bits 31 to 8 of EIPSW are reserved (fixed to 0) for future function expansion.

When the RETI instruction is executed, the values in EIPC and EIPSW are restored to the PC and PSW, respectively.



(2) NMI status saving registers (FEPC, FEPSW)

There are two NMI status saving registers, FEPC and FEPSW.

Upon occurrence of a non-maskable interrupt (NMI), the contents of the program counter (PC) are saved to FEPC and the contents of the program status word (PSW) are saved to FEPSW.

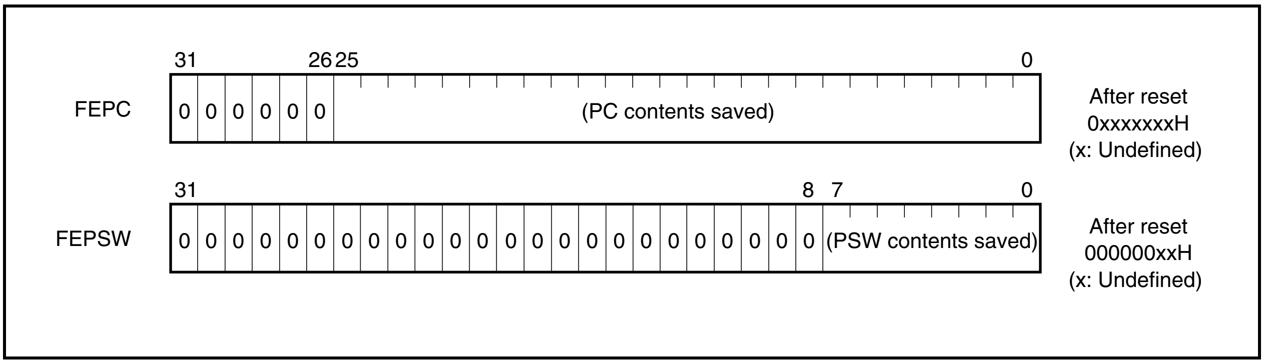
The address of the next instruction following the instruction executed when a non-maskable interrupt occurs is saved to FEPC, except for some instructions.

The current PSW contents are saved to FEPSW.

Since there is only one set of NMI status saving registers, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.

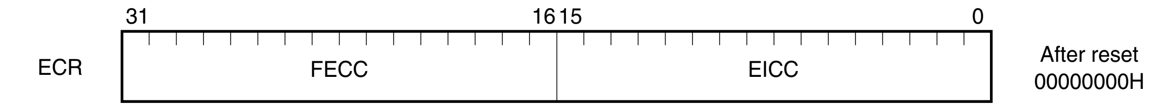
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved (fixed to 0) for future function expansion.

When the RETI instruction has been executed, the values of FEPC and FEPSW are restored to the PC and PSW, respectively.



(3) Interrupt source register (ECR)

Upon occurrence of an interrupt or an exception, the interrupt source register (ECR) holds the source of an interrupt or an exception. The value held by ECR is the exception code coded for each interrupt source. This register is a read-only register, and thus data cannot be written to it using the LDSR instruction.



Bit position	Bit name	Description
31 to 16	FECC	Non-maskable interrupt (NMI) exception code
15 to 0	EICC	Exception, maskable interrupt exception code

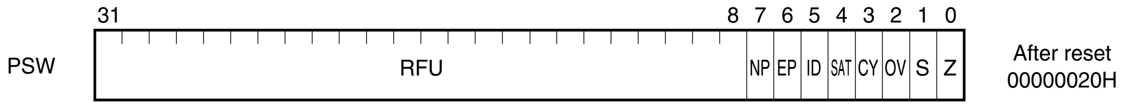
(4) Program status word (PSW)

The program status word (PSW) is a collection of flags that indicate the program status (instruction execution result) and the CPU status.

When the contents of this register are changed using the LDSR instruction, the new contents become valid immediately following completion of LDSR instruction execution. Interrupt request acknowledgment is held pending while a write to the PSW is being executed by the LDSR instruction.

Bits 31 to 8 are reserved (fixed to 0) for future function expansion.

(1/2)



Bit position	Flag name	Description
31 to 8	RFU	Reserved field. Fixed to 0.
7	NP	Indicates that non-maskable interrupt (NMI) servicing is in progress. This flag is set to 1 when an NMI request is acknowledged, and disables multiple interrupts. 0: NMI servicing not in progress 1: NMI servicing in progress
6	EP	Indicates that exception processing is in progress. This flag is set to 1 when an exception occurs. Moreover, interrupt requests can be acknowledged even when this bit is set. 0: Exception processing not in progress 1: Exception processing in progress
5	ID	Indicates whether maskable interrupt request acknowledgment is enabled. 0: Interrupt enabled (EI) 1: Interrupt disabled (DI)
4	SAT[Note]	Indicates that the result of executing a saturated operation instruction has overflowed and that the calculation result is saturated. Since this is a cumulative flag, it is set to 1 when the result of a saturated operation instruction becomes saturated, and it is not cleared to 0 even if the operation results of successive instructions do not become saturated. This flag is neither set nor cleared when arithmetic operation instructions are executed. 0: Not saturated 1: Saturated
3	CY	Indicates whether carry or borrow occurred as the result of an operation. 0: No carry or borrow occurred 1: Carry or borrow occurred
2	OV[Note]	Indicates whether overflow occurred during an operation. 0: No overflow occurred 1: Overflow occurred.
1	S[Note]	Indicates whether the result of an operation is negative. 0: Operation result is positive or 0. 1: Operation result is negative.
0	Z	Indicates whether operation result is 0. 0: Operation result is not 0. 1: Operation result is 0.

Remark **Note** is explained on the following page.

(2/2)

Note During saturated operation, the saturated operation results are determined by the contents of the OV flag and S flag. The SAT flag is set (to 1) only when the OV flag is set (to 1) during saturated operation.

Operation result status	Flag status			Saturated operation result
	SAT	OV	S	
Maximum positive value exceeded	1	1	0	7FFFFFFFH
Maximum negative value exceeded	1	1	1	80000000H
Positive (maximum value not exceeded)	Holds value before operation	0	0	Actual operation result
Negative (maximum value not exceeded)			1	

(5) CALLT execution status saving registers (CTPC, CTPSW)

There are two CALLT execution status saving registers, CTPC and CTPSW.

When the CALLT instruction is executed, the contents of the program counter (PC) are saved to CTPC, and the program status word (PSW) contents are saved to CTPSW.

The contents saved to CTPC consist of the address of the next instruction after the CALLT instruction.

The current PSW contents are saved to CTPSW.

Bits 31 to 26 of CTPC and bits 31 to 8 of CTPSW are reserved (fixed to 0) for future function expansion.

	31					26	25 0	
CTPC	0	0	0	0	0	0	(PC contents saved)	After reset 0xxxxxxxH (x: Undefined)

	31																							8	7 0	
CTPSW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(PSW contents saved)	After reset 000000xxH (x: Undefined)

(6) Exception/debug trap status saving registers (DBPC, DBPSW)

There are two exception/debug trap status saving registers, DBPC and DBPSW.

Upon occurrence of an exception trap or debug trap, the contents of the program counter (PC) are saved to DBPC, and the program status word (PSW) contents are saved to DBPSW.

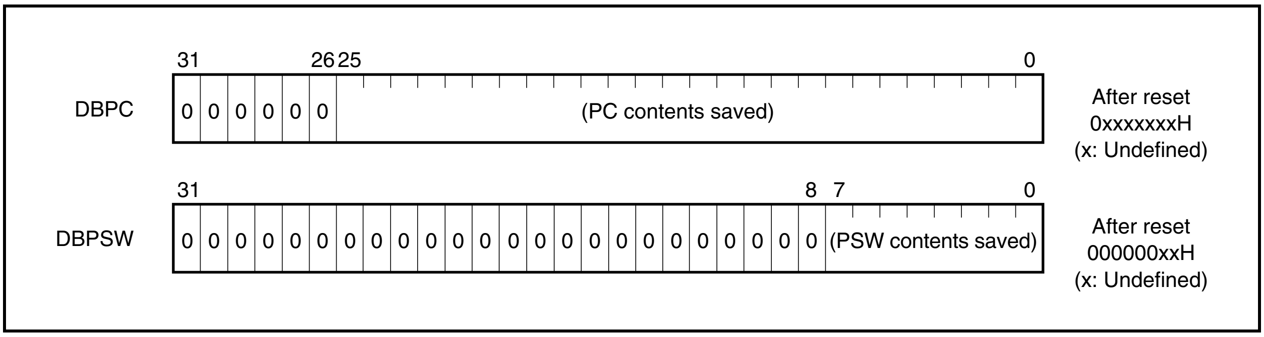
The contents saved to DBPC consist of the address of the next instruction after the instruction executed when an exception trap or debug trap occurs.

The current PSW contents are saved to DBPSW.

These registers can be read or written only in the period between DBTRAP instruction or illegal opcode execution and DBRET instruction execution.

Bits 31 to 26 of DBPC and bits 31 to 8 of DBPSW are reserved (fixed to 0) for future function expansion.

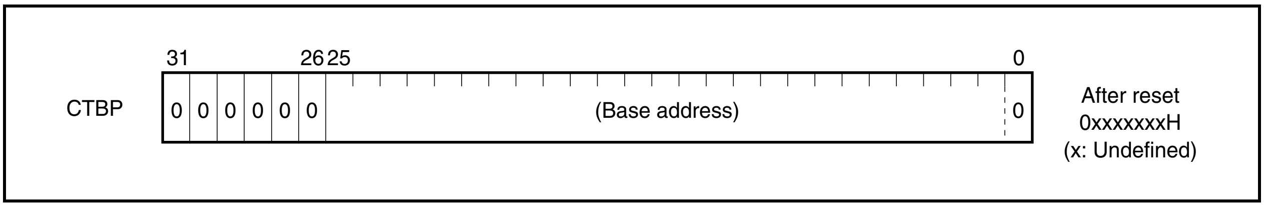
When the DBRET instruction has been executed, the values of DBPC and DBPSW are restored to the PC and PSW, respectively.



(7) CALLT base pointer (CTBP)

The CALLT base pointer (CTBP) is used to specify table addresses and generate target addresses (bit 0 is fixed to 0).

Bits 31 to 26 are reserved (fixed to 0) for future function expansion.



3.3 Operating Modes

3.3.1 Operating modes

The V850E/IG4-H and V850E/IH4-H have the following operating modes. Mode specification is carried out using the FLMD0 and FLMD1 pins.

(1) Normal operation mode

In this mode, execution branches to the reset entry address in the internal ROM and instruction processing is started when system reset is released.

(2) Flash memory programming mode

If this mode is specified, a program can be written to the internal flash memory by the flash memory programmer.

3.3.2 Operating mode specification

The operating mode is specified according to the status (input level) of the FLMD0 and FLMD1 pins.

In the normal operating mode, input a low level to the FLMD0 pin after reset.

When the flash memory programmer is connected, a high level is input to the FLMD0 pin by the flash memory programmer in the flash memory programming mode; however, in the self-programming mode, input a high level via an external circuit.

Other than in the self-programming mode, fix the specifications of these pins in the application system, and do not change then during operation.

FLMD1	FLMD0	Operating Mode	Remarks
×	L	Normal operation mode	Internal ROM area is allocated from address 000000H.
L	H	Flash memory programming mode	–
H	H	Setting prohibited	

Remark L: Low-level input
H: High-level input

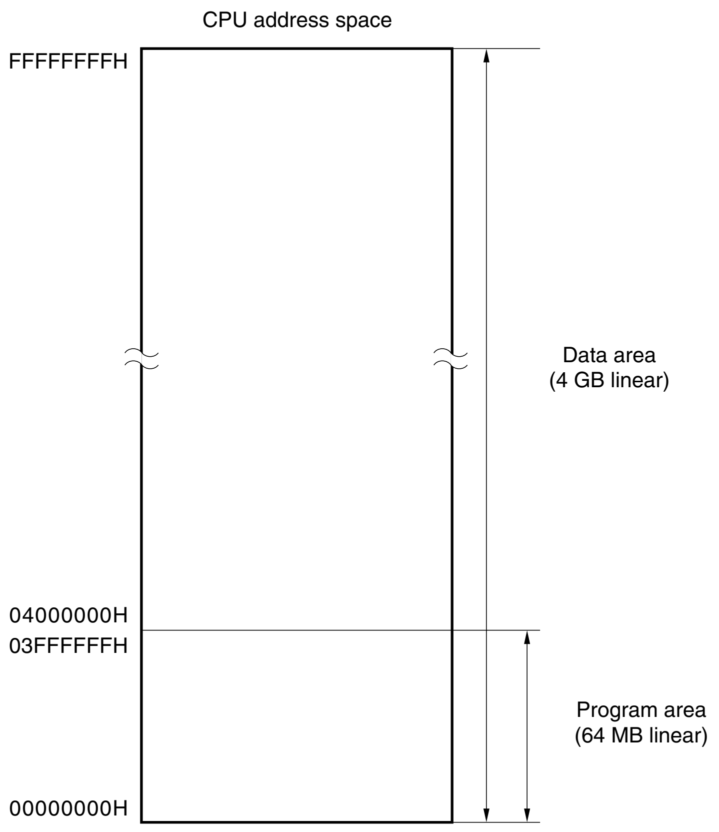
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850E/IG4-H and V850E/IH4-H has 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). Also, in instruction address addressing, a maximum of 64 MB of linear address space (program space) is supported.

Figure 3-2 shows the CPU address space.

Figure 3-2. CPU Address Space

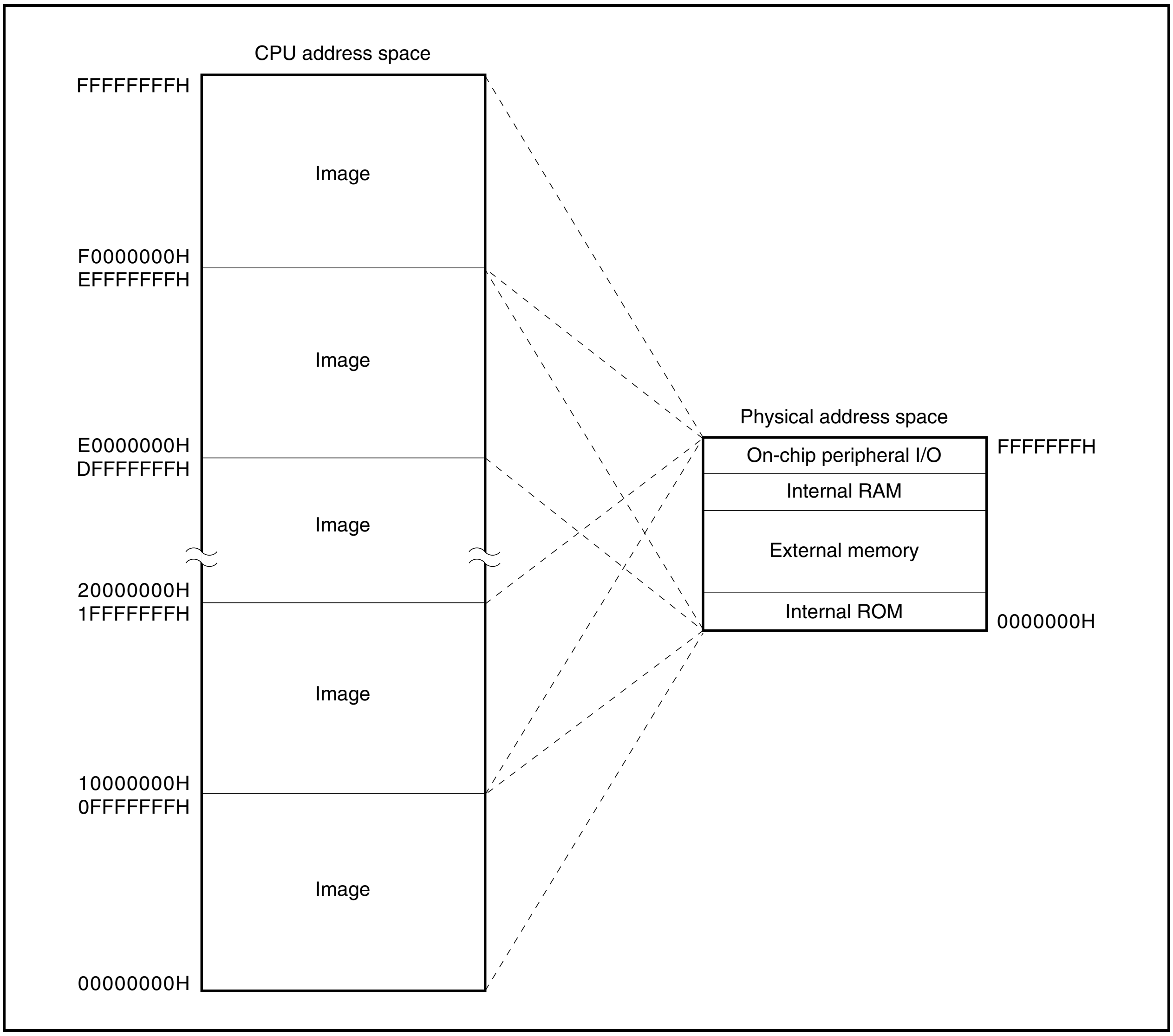


3.4.2 Image

A 256 MB physical address space is seen as 16 images in the 4 GB CPU address space. In actuality, the same 256 MB physical address space is accessed regardless of the values of bits 31 to 28 of the CPU address. Figure 3-3 shows the image of the virtual addressing space.

Physical address x0000000H can be seen as CPU address 00000000H, and in addition, can be seen as address 10000000H, address 20000000H, … , address E0000000H, or address F0000000H.

Figure 3-3. Images on Address Space



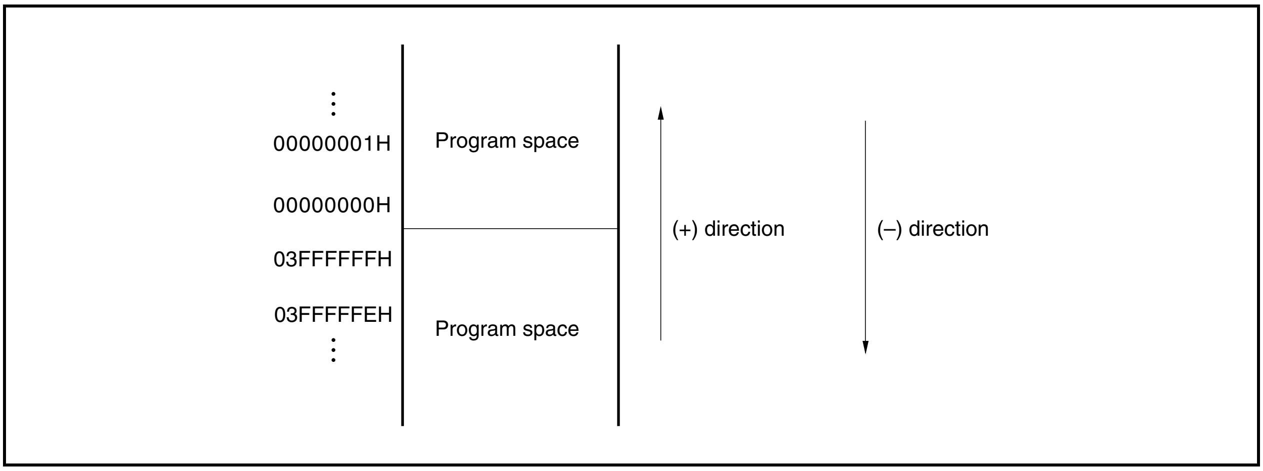
3.4.3 Wraparound of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Even if a carry or borrow occurs from bit 25 to 26 as a result of a branch address calculation, the higher 6 bits ignore the carry or borrow.

Therefore, the upper-limit address of the program space, address 03FFFFFFH, and the lower-limit address 00000000H become contiguous addresses. Wraparound refers to a situation like this whereby the lower-limit address and upper-limit address become contiguous.

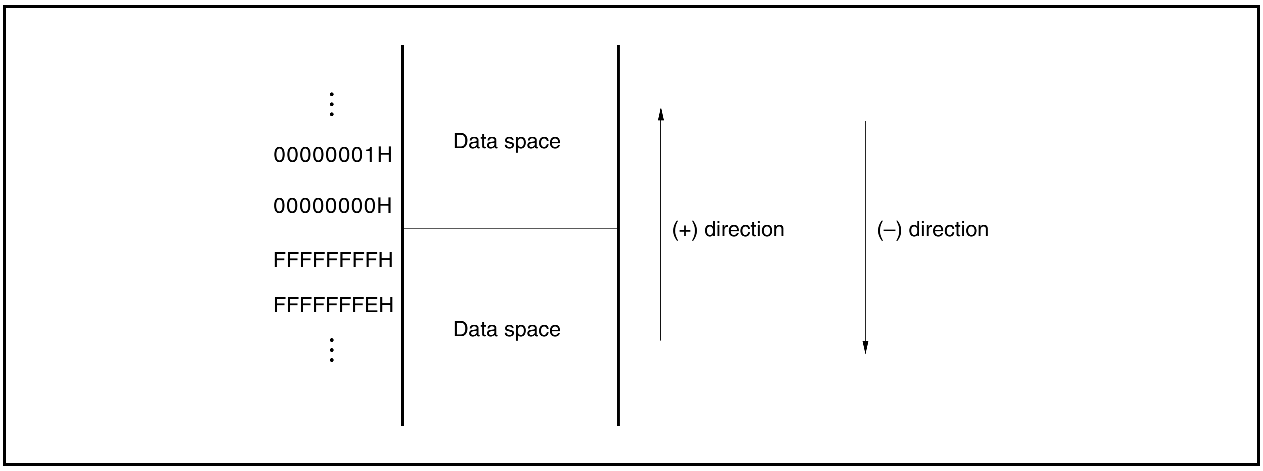
Caution The 4 KB area of 03FFF000H to 03FFFFFFH can be seen as an image of 0FFFF000H to 0FFFFFFFH. This area is access-prohibited. Therefore, do not execute any branch address calculation in which the result will reside in any part of this area.



(2) Data space

The result of an operand address calculation that exceeds 32 bits is ignored.

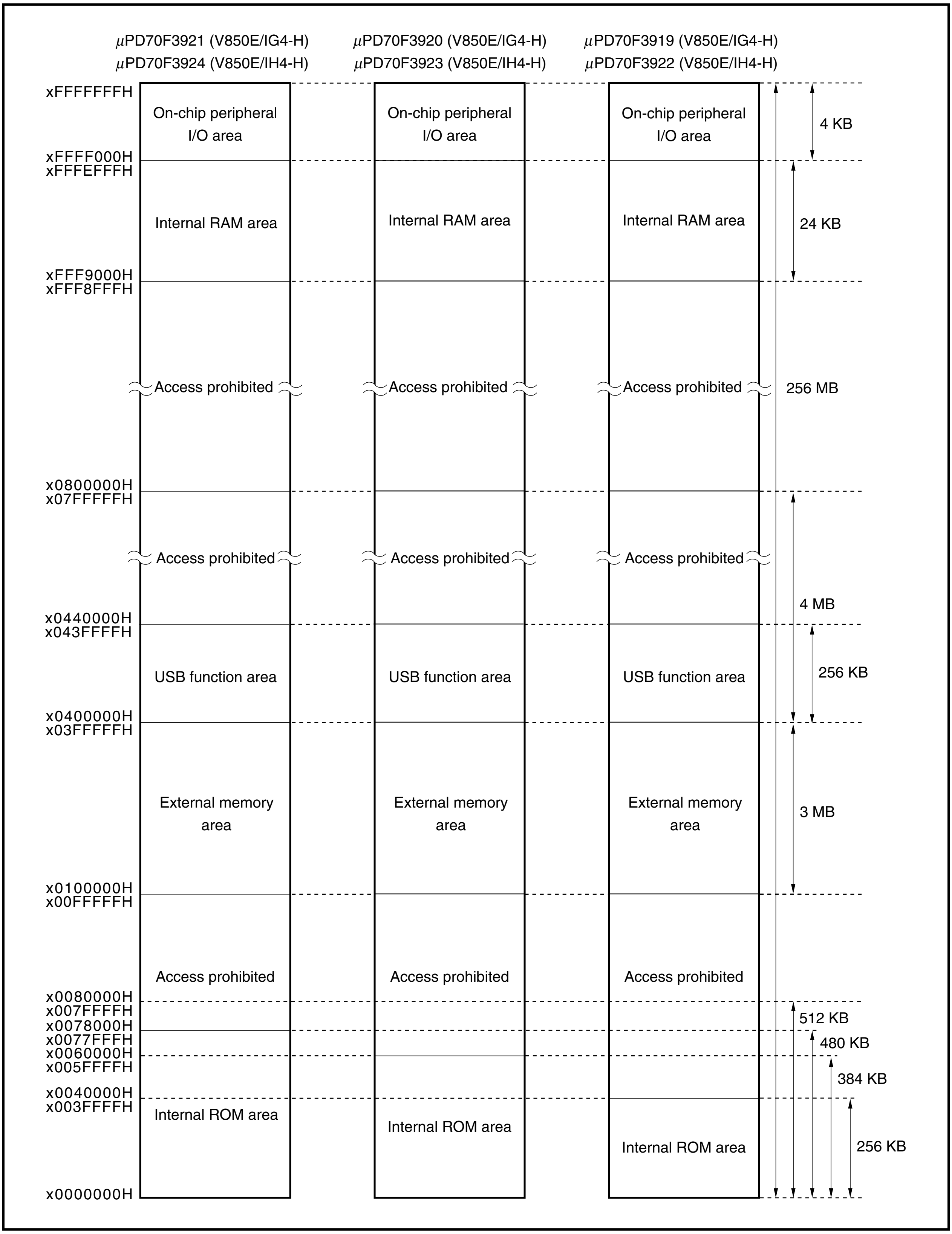
Therefore, the upper-limit address of the program space, address FFFFFFFFH, and the lower-limit address 00000000H are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.



3.4.4 Memory map

The V850E/IG4-H and V850E/IH4-H reserve areas as shown in Figure 3-4.

Figure 3-4. Memory Map



3.4.5 Areas

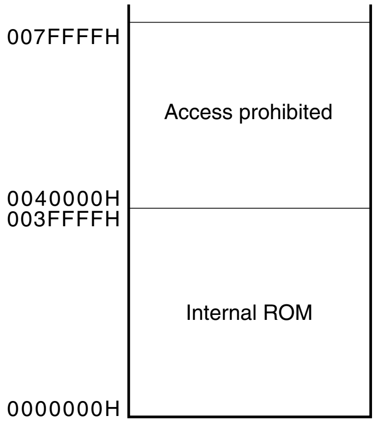
(1) Internal ROM area

512 KB of internal ROM area, addresses 00000H to 7FFFFH, is reserved.

(a) μPD70F3919 (V850E/IG4-H), μPD70F3922 (V850E/IH4-H)

256 KB are provided at addresses 000000H to 03FFFFH as physical internal ROM.

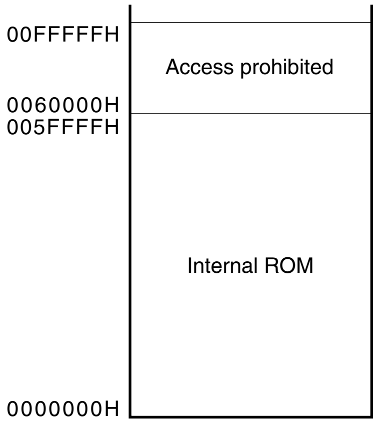
Figure 3-5. Internal ROM Area (256 KB)



(b) μPD70F3920 (V850E/IG4-H), μPD70F3923 (V850E/IH4-H)

384 KB are provided at addresses 000000H to 05FFFFH as physical internal ROM.

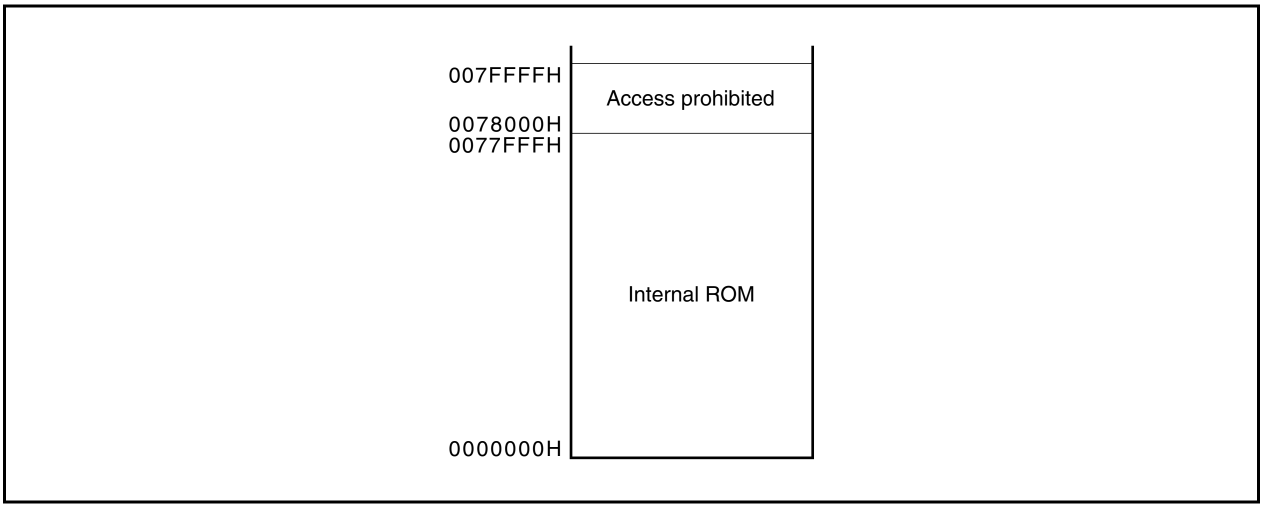
Figure 3-6. Internal ROM Area (384 KB)



(c) μPD70F3921 (V850E/IG4-H), μPD70F3924 (V850E/IH4-H)

480 KB are provided at addresses 000000H to 077FFFH as physical internal ROM.

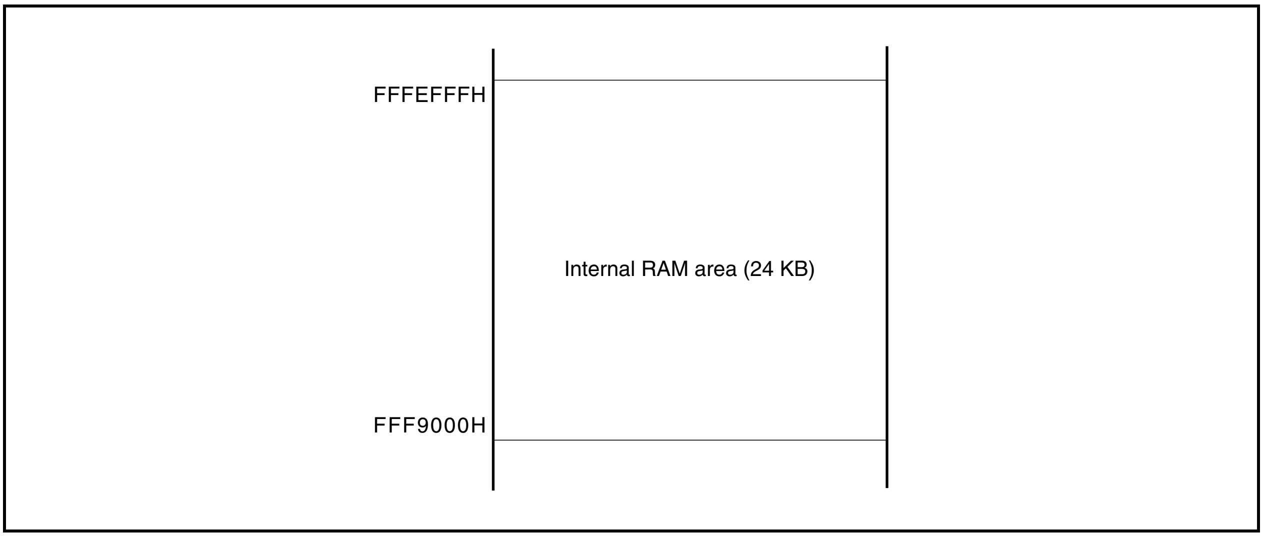
Figure 3-7. Internal ROM Area (480 KB)



(2) Internal RAM area

24 KB are provided at addresses FFF9000H to FFFEFFFH as physical internal RAM.

Figure 3-8. Internal RAM Area (24 KB)

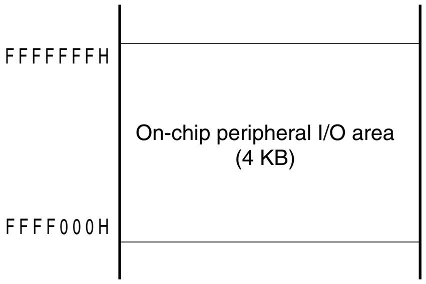


(3) On-chip peripheral I/O area

4 KB of memory, addresses FFFF000H to FFFFFFFH, is provided as an on-chip peripheral I/O area.
An image of addresses FFFF000H to FFFFFFFH can be seen at addresses 3FFF000H to 3FFFFFFH[Note].

Note Addresses 3FFF000H to 3FFFFFFH are access-prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.

Figure 3-9. On-Chip Peripheral I/O Area



On-chip peripheral I/O registers associated with the operating mode specification and the state monitoring for the on-chip peripheral I/O are all memory-mapped to the on-chip peripheral I/O area. Program fetches cannot be executed from this area.

Cautions 1. In the V850E/IG4-H and V850E/IH4-H, if a register is word accessed, halfword access is performed twice in the order of lower address, then higher address of the word area, disregarding the lower 2 bits of the address.

2. For registers in which byte access is possible, if halfword access is executed, the higher 8 bits become undefined during the read operation, and the lower 8 bits of data are written to the register during the write operation.

3. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.
Addresses 3FFF000H to 3FFFFFFH cannot be specified as the source/destination address of DMA transfer. Be sure to use addresses FFFF000H to FFFFFFFH for the source/destination address of DMA transfer.

(4) External memory area

3 MB (0100000H to 03FFFFFH) are available for the external memory area. For details, see **CHAPTER 19 BUS CONTROL FUNCTION**.

3.4.6 Recommended use of address space

The architecture of the V850E/IG4-H and V850E/IH4-H requires that a register that serves as a pointer be secured for address generation in operand data accessing of data space. Operand data access from instruction can be directly executed at the address in this pointer register area ±32 KB. However, because the general-purpose registers that can be used as a pointer register are limited, by minimizing the deterioration of address calculation performance when changing the pointer value, the number of usable general-purpose registers for handling variables is maximized, and the program size can be saved.

(1) Program space

Of the 32 bits of the program counter (PC), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Therefore, a contiguous 64 MB space, starting from address 00000000H, unconditionally corresponds to the memory map of the program space.

(2) Data space

With the V850E/IG4-H and V850E/IH4-H, a 256 MB physical address space is seen as 16 images in the 4 GB CPU address space. The highest bit (bit 25) of this 26-bit address is assigned as an address sign-extended to 32 bits.

(a) Application examples using wraparound

When R = r0 (zero register) is specified by the LD/ST disp16 [R] instruction, an addressing range of 00000000H ±32 KB can be referenced by the sign-extended disp16.

The zero register (r0) is a register set to 0 by the hardware, and eliminates the need for additional registers for the pointer.

Example μPD70F3919 (V850E/IG4-H)

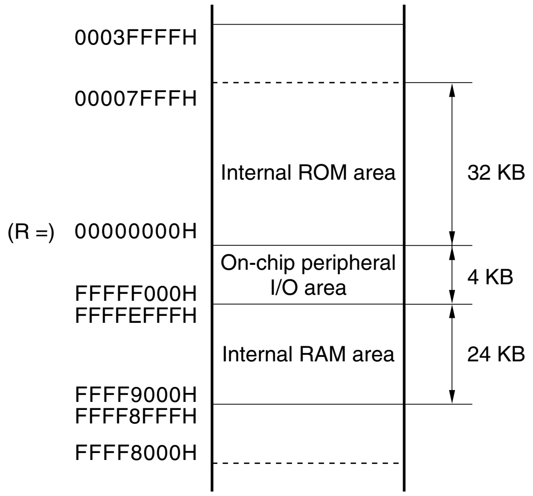
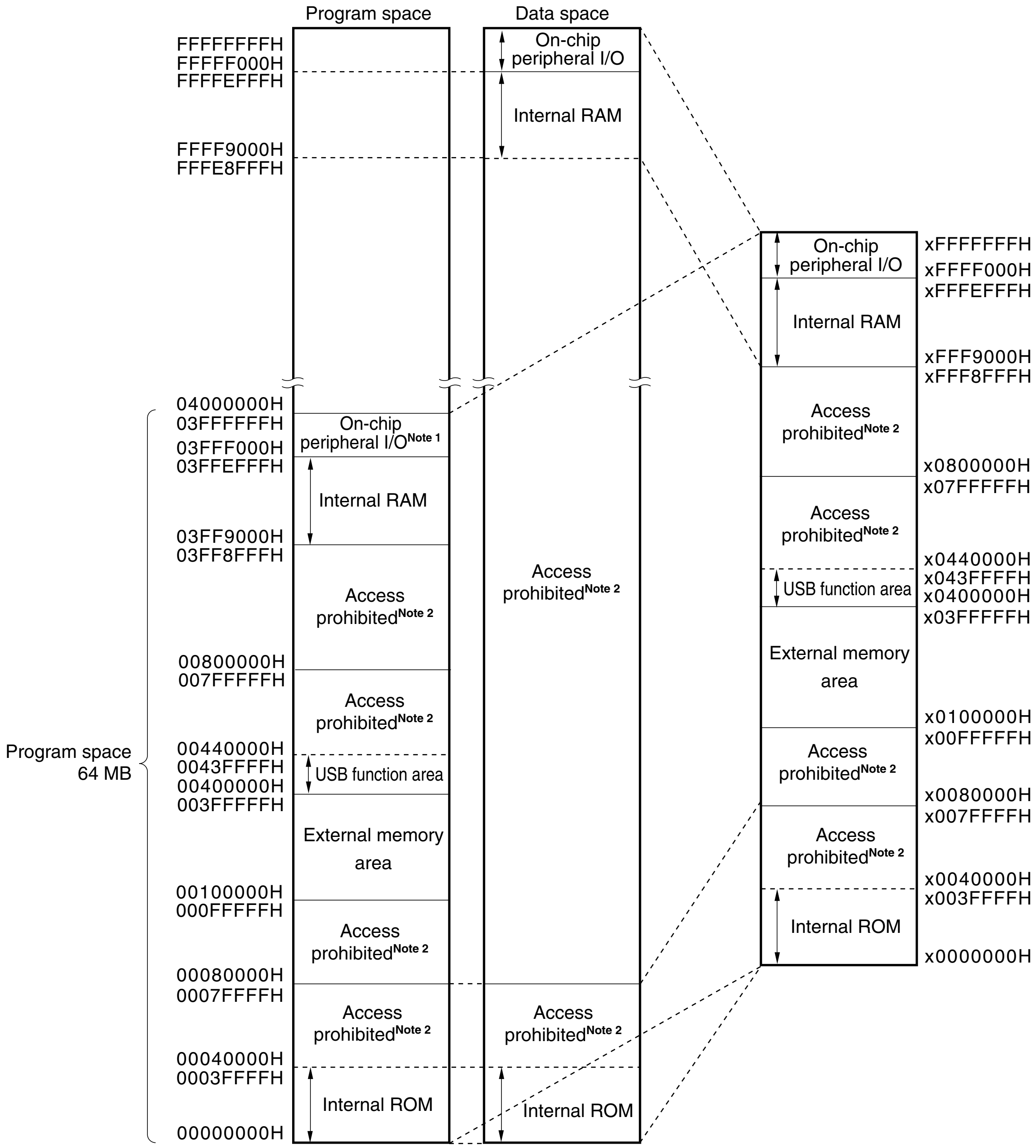


Figure 3-10. Recommended Memory Map



Notes 1. This area is access-prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.

2. The operation is not guaranteed if an access-prohibited area is accessed.

Remarks 1. The arrows indicate the recommended area.

2. This is a recommended memory map for the μPD70F3919 (V850E/IG4-H).

3.4.7 On-chip peripheral I/O registers

(1/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1	8	16	32	
FFFFF004H	Port DL register	PDL	R/W			√		Undefined
FFFFF004H	Port DLL register	PDLL		√	√			Undefined
FFFFF005H	Port DLH register	PDLH		√	√			Undefined
FFFFF024H	Port DL mode register	PMDL				√		FFFFH
FFFFF024H	Port DL mode register L	PMDLL		√	√			FFH
FFFFF025H	Port DL mode register H	PMDLH		√	√			FFH
FFFFF044H	Port DL mode control register	PMCDL				√		0000H
FFFFF044H	Port DL mode control register L	PMCDLL		√	√			00H
FFFFF045H	Port DL mode control register H	PMCDLH		√	√			00H
FFFFF066H	Bus size configuration register	BSC				√		5555H
FFFFF06EH	System wait control register	VSWC			√			77H
FFFFF080H	DMA trigger factor register 0	DTFR0				√		0000H
FFFFF080H	DMA trigger factor register 0L	DTFR0L			√			00H
FFFFF081H	DMA trigger factor register 0H	DTFR0H			√			00H
FFFFF082H	DMA addressing control register 0	DADC0				√		0000H
FFFFF084H	DMA transfer count specification register 0	DTCR0				√		Undefined
FFFFF086H	DMA transfer destination address specification register 0	DDAR0					√	Undefined
FFFFF086H	DMA transfer destination address specification register 0L	DDAR0L				√		Undefined
FFFFF088H	DMA transfer destination address specification register 0H	DDAR0H				√		Undefined
FFFFF08AH	DMA transfer source address specification register 0	DSAR0					√	Undefined
FFFFF08AH	DMA transfer source address specification register 0L	DSAR0L				√		Undefined
FFFFF08CH	DMA transfer source address specification register 0H	DSAR0H				√		Undefined
FFFFF08EH	DMA channel control register 0	DCHC0				√		0000H
FFFFF090H	DMA trigger factor register 1	DTFR1				√		0000H
FFFFF090H	DMA trigger factor register 1L	DTFR1L			√			00H
FFFFF091H	DMA trigger factor register 1H	DTFR1H			√			00H
FFFFF092H	DMA addressing control register 1	DADC1				√		0000H
FFFFF094H	DMA transfer count specification register 1	DTCR1				√		Undefined
FFFFF096H	DMA transfer destination address specification register 1	DDAR1					√	Undefined
FFFFF096H	DMA transfer destination address specification register 1L	DDAR1L				√		Undefined
FFFFF098H	DMA transfer destination address specification register 1H	DDAR1H				√		Undefined
FFFFF09AH	DMA transfer source address specification register 1	DSAR1					√	Undefined
FFFFF09AH	DMA transfer source address specification register 1L	DSAR1L				√		Undefined
FFFFF09CH	DMA transfer source address specification register 1H	DSAR1H				√		Undefined
FFFFF09EH	DMA channel control register 1	DCHC1				√		0000H

(2/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1	8	16	32	
FFFFF0A0H	DMA trigger factor register 2	DTFR2	R/W			√		0000H
FFFFF0A0H	DMA trigger factor register 2L	DTFR2L			√			00H
FFFFF0A1H	DMA trigger factor register 2H	DTFR2H			√			00H
FFFFF0A2H	DMA addressing control register 2	DADC2				√		0000H
FFFFF0A4H	DMA transfer count specification register 2	DTCR2				√		Undefined
FFFFF0A6H	DMA transfer destination address specification register 2	DDAR2					√	Undefined
FFFFF0A6H	DMA transfer destination address specification register 2L	DDAR2L				√		Undefined
FFFFF0A8H	DMA transfer destination address specification register 2H	DDAR2H				√		Undefined
FFFFF0AAH	DMA transfer source address specification register 2	DSAR2					√	Undefined
FFFFF0AAH	DMA transfer source address specification register 2L	DSAR2L				√		Undefined
FFFFF0ACH	DMA transfer source address specification register 2H	DSAR2H				√		Undefined
FFFFF0AEH	DMA channel control register 2	DCHC2				√		0000H
FFFFF0B0H	DMA trigger factor register 3	DTFR3				√		0000H
FFFFF0B0H	DMA trigger factor register 3L	DTFR3L			√			00H
FFFFF0B1H	DMA trigger factor register 3H	DTFR3H			√			00H
FFFFF0B2H	DMA addressing control register 3	DADC3				√		0000H
FFFFF0B4H	DMA transfer count specification register 3	DTCR3				√		Undefined
FFFFF0B6H	DMA transfer destination address specification register 3	DDAR3					√	Undefined
FFFFF0B6H	DMA transfer destination address specification register 3L	DDAR3L				√		Undefined
FFFFF0B8H	DMA transfer destination address specification register 3H	DDAR3H				√		Undefined
FFFFF0BAH	DMA transfer source address specification register 3	DSAR3					√	Undefined
FFFFF0BAH	DMA transfer source address specification register 3L	DSAR3L				√		Undefined
FFFFF0BCH	DMA transfer source address specification register 3H	DSAR3H				√		Undefined
FFFFF0BEH	DMA channel control register 3	DCHC3				√		0000H
FFFFF0C0H	DMA trigger factor register 4	DTFR4				√		0000H
FFFFF0C0H	DMA trigger factor register 4L	DTFR4L			√			00H
FFFFF0C1H	DMA trigger factor register 4H	DTFR4H			√			00H
FFFFF0C2H	DMA addressing control register 4	DADC4				√		0000H
FFFFF0C4H	DMA transfer count specification register 4	DTCR4				√		Undefined
FFFFF0C6H	DMA transfer destination address specification register 4	DDAR4					√	Undefined
FFFFF0C6H	DMA transfer destination address specification register 4L	DDAR4L				√		Undefined
FFFFF0C8H	DMA transfer destination address specification register 4H	DDAR4H				√		Undefined

(3/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1	8	16	32	
FFFFF0CAH	DMA transfer source address specification register 4	DSAR4	R/W				√	Undefined
FFFFF0CAH	DMA transfer source address specification register 4L	DSAR4L				√		Undefined
FFFFF0CCH	DMA transfer source address specification register 4H	DSAR4H				√		Undefined
FFFFF0CEH	DMA channel control register 4	DCHC4				√		0000H
FFFFF0D0H	DMA trigger factor register 5	DTFR5				√		0000H
FFFFF0D0H	DMA trigger factor register 5L	DTFR5L			√			00H
FFFFF0D1H	DMA trigger factor register 5H	DTFR5H			√			00H
FFFFF0D2H	DMA addressing control register 5	DADC5				√		0000H
FFFFF0D4H	DMA transfer count specification register 5	DTCR5				√		Undefined
FFFFF0D6H	DMA transfer destination address specification register 5	DDAR5					√	Undefined
FFFFF0D6H	DMA transfer destination address specification register 5L	DDAR5L				√		Undefined
FFFFF0D8H	DMA transfer destination address specification register 5H	DDAR5H				√		Undefined
FFFFF0DAH	DMA transfer source address specification register 5	DDAR5					√	Undefined
FFFFF0DAH	DMA transfer source address specification register 5L	DDAR5L				√		Undefined
FFFFF0DCH	DMA transfer source address specification register 5H	DDAR5H				√		Undefined
FFFFF0DEH	DMA channel control register 5	DCHC5				√		0000H
FFFFF0E0H	DMA trigger factor register 6	DTFR6				√		0000H
FFFFF0E0H	DMA trigger factor register 6L	DTFR6L			√			00H
FFFFF0E1H	DMA trigger factor register 6H	DTFR6H			√			00H
FFFFF0E2H	DMA addressing control register 6	DADC6				√		0000H
FFFFF0E4H	DMA transfer count specification register 6	DTCR6				√		Undefined
FFFFF0E6H	DMA transfer destination address specification register 6	DDAR6					√	Undefined
FFFFF0E6H	DMA transfer destination address specification register 6L	DDAR6L				√		Undefined
FFFFF0E8H	DMA transfer destination address specification register 6H	DDAR6H				√		Undefined
FFFFF0EAH	DMA transfer source address specification register 6	DSAR6					√	Undefined
FFFFF0EAH	DMA transfer source address specification register 6L	DSAR6L				√		Undefined
FFFFF0ECH	DMA transfer source address specification register 6H	DSAR6H				√		Undefined
FFFFF0EEH	DMA channel control register 6	DCHC6				√		0000H
FFFFF0F0H	DMA status register	DMAS		√	√			00H
FFFFF0F2H	DMA enable register	DEN			√			00H
FFFFF0F4H	DMA stop register	DMSTP			√			00H
FFFFF100H	Interrupt mask register 0	IMR0				√		FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√			FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√			FFH

(4/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF102H	Interrupt mask register 1	IMR1	R/W			√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2				√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H		√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3				√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L		√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H		√	√		FFH
FFFFF108H	Interrupt mask register 4	IMR4				√	FFFFH
FFFFF108H	Interrupt mask register 4L	IMR4L		√	√		FFH
FFFFF109H	Interrupt mask register 4H	IMR4H		√	√		FFH
FFFFF10AH	Interrupt mask register 5	IMR5				√	FFFFH
FFFFF10AH	Interrupt mask register 5L	IMR5L		√	√		FFH
FFFFF10BH	Interrupt mask register 5H	IMR5H		√	√		FFH
FFFFF10CH	Interrupt mask register 6	IMR6				√	FFFFH
FFFFF10CH	Interrupt mask register 6L	IMR6L		√	√		FFH
FFFFF10DH	Interrupt mask register 6H	IMR6H		√	√		FFH
FFFFF110H	Interrupt control register	LVILIC		√	√		47H
FFFFF112H	Interrupt control register	LVIHIC		√	√		47H
FFFFF114H	Interrupt control register	PIC00		√	√		47H
FFFFF116H	Interrupt control register	PIC01		√	√		47H
FFFFF118H	Interrupt control register	PIC02		√	√		47H
FFFFF11AH	Interrupt control register	PIC03		√	√		47H
FFFFF11CH	Interrupt control register	PIC04		√	√		47H
FFFFF11EH	Interrupt control register	PIC05		√	√		47H
FFFFF120H	Interrupt control register	PIC06		√	√		47H
FFFFF122H	Interrupt control register	PIC07		√	√		47H
FFFFF124H	Interrupt control register	PIC08		√	√		47H
FFFFF126H	Interrupt control register	PIC09		√	√		47H
FFFFF128H	Interrupt control register	PIC10		√	√		47H
FFFFF12AH	Interrupt control register	PIC11		√	√		47H
FFFFF12CH	Interrupt control register	PIC12		√	√		47H
FFFFF12EH	Interrupt control register	PIC13		√	√		47H
FFFFF130H	Interrupt control register	PIC14		√	√		47H
FFFFF132H	Interrupt control register	PIC15		√	√		47H
FFFFF134H	Interrupt control register	PIC16		√	√		47H
FFFFF136H	Interrupt control register	PIC17		√	√		47H
FFFFF138H	Interrupt control register	PIC18		√	√		47H
FFFFF13AH	Interrupt control register	PIC19		√	√		47H
FFFFF13CH	Interrupt control register	CMPIC0L		√	√		47H

(5/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF13EH	Interrupt control register	CMPIC0F	R/W	√	√		47H
FFFFF140H	Interrupt control register	CMPIC1L		√	√		47H
FFFFF142H	Interrupt control register	CMPIC1F		√	√		47H
FFFFF144H	Interrupt control register	TB0OVIC		√	√		47H
FFFFF146H	Interrupt control register	TB0CCIC0		√	√		47H
FFFFF148H	Interrupt control register	TB0CCIC1		√	√		47H
FFFFF14AH	Interrupt control register	TB0CCIC2		√	√		47H
FFFFF14CH	Interrupt control register	TB0CCIC3		√	√		47H
FFFFF14EH	Interrupt control register	TB1OVIC		√	√		47H
FFFFF150H	Interrupt control register	TB1CCIC0		√	√		47H
FFFFF152H	Interrupt control register	TB1CCIC1		√	√		47H
FFFFF154H	Interrupt control register	TB1CCIC2		√	√		47H
FFFFF156H	Interrupt control register	TB1CCIC3		√	√		47H
FFFFF158H	Interrupt control register	TT0OVIC		√	√		47H
FFFFF15AH	Interrupt control register	TT0CCIC0		√	√		47H
FFFFF15CH	Interrupt control register	TT0CCIC1		√	√		47H
FFFFF15EH	Interrupt control register	TT0IECIC		√	√		47H
FFFFF160H	Interrupt control register	TT1OVIC		√	√		47H
FFFFF162H	Interrupt control register	TT1CCIC0		√	√		47H
FFFFF164H	Interrupt control register	TT1CCIC1		√	√		47H
FFFFF166H	Interrupt control register	TT1IECIC		√	√		47H
FFFFF168H	Interrupt control register	TT2OVIC		√	√		47H
FFFFF16AH	Interrupt control register	TT2CCIC0		√	√		47H
FFFFF16CH	Interrupt control register	TT2CCIC1		√	√		47H
FFFFF16EH	Interrupt control register	TT3OVIC		√	√		47H
FFFFF170H	Interrupt control register	TT3CCIC0		√	√		47H
FFFFF172H	Interrupt control register	TT3CCIC1		√	√		47H
FFFFF174H	Interrupt control register	TA0OVIC		√	√		47H
FFFFF176H	Interrupt control register	TA0CCIC0		√	√		47H
FFFFF178H	Interrupt control register	TA0CCIC1		√	√		47H
FFFFF17AH	Interrupt control register	TA1OVIC		√	√		47H
FFFFF17CH	Interrupt control register	TA1CCIC0		√	√		47H
FFFFF17EH	Interrupt control register	TA1CCIC1		√	√		47H
FFFFF180H	Interrupt control register	TA2OVIC		√	√		47H
FFFFF182H	Interrupt control register	TA2CCIC0		√	√		47H
FFFFF184H	Interrupt control register	TA2CCIC1		√	√		47H
FFFFF186H	Interrupt control register	DMAIC0		√	√		47H
FFFFF188H	Interrupt control register	DMAIC1		√	√		47H
FFFFF18AH	Interrupt control register	DMAIC2		√	√		47H
FFFFF18CH	Interrupt control register	DMAIC3		√	√		47H
FFFFF18EH	Interrupt control register	DMAIC4		√	√		47H

(6/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF190H	Interrupt control register	DMAIC5	R/W	√	√		47H
FFFFF192H	Interrupt control register	UREIC		√	√		47H
FFFFF194H	Interrupt control register	URIC		√	√		47H
FFFFF196H	Interrupt control register	UTIC		√	√		47H
FFFFF198H	Interrupt control register	UIFIC		√	√		47H
FFFFF19AH	Interrupt control register	UTOIC		√	√		47H
FFFFF19CH	Interrupt control register	UA0REIC		√	√		47H
FFFFF19EH	Interrupt control register	UA0RIC		√	√		47H
FFFFF1A0H	Interrupt control register	UA0TIC		√	√		47H
FFFFF1A2H	Interrupt control register	CF0REIC		√	√		47H
FFFFF1A4H	Interrupt control register	CF0RIC		√	√		47H
FFFFF1A6H	Interrupt control register	CF0TIC		√	√		47H
FFFFF1A8H	Interrupt control register	UA1REIC		√	√		47H
FFFFF1AAH	Interrupt control register	UA1RIC		√	√		47H
FFFFF1ACH	Interrupt control register	UA1TIC		√	√		47H
FFFFF1AEH	Interrupt control register	CF1REIC		√	√		47H
FFFFF1B0H	Interrupt control register	CF1RIC		√	√		47H
FFFFF1B2H	Interrupt control register	CF1TIC		√	√		47H
FFFFF1B4H	Interrupt control register	UA2REIC		√	√		47H
FFFFF1B6H	Interrupt control register	UA2RIC		√	√		47H
FFFFF1B8H	Interrupt control register	UA2TIC		√	√		47H
FFFFF1BAH	Interrupt control register	CF2REIC		√	√		47H
FFFFF1BCH	Interrupt control register	CF2RIC		√	√		47H
FFFFF1BEH	Interrupt control register	CF2TIC		√	√		47H
FFFFF1C0H	Interrupt control register	IICIC		√	√		47H
FFFFF1C2H	Interrupt control register	AD0IC		√	√		47H
FFFFF1C4H	Interrupt control register	AD1IC		√	√		47H
FFFFF1C6H	Interrupt control register	AD2IC		√	√		47H
FFFFF1C8H	Interrupt control register	TM0EQIC0		√	√		47H
FFFFF1CAH	Interrupt control register	TM1EQIC0		√	√		47H
FFFFF1CCH	Interrupt control register	TM2EQIC0		√	√		47H
FFFFF1CEH	Interrupt control register	TM3EQIC0		√	√		47H
FFFFF1D0H	Interrupt control register	ADT0IC		√	√		47H
FFFFF1D2H	Interrupt control register	ADT1IC		√	√		47H
FFFFF1D4H	Interrupt control register	UFIC0		√	√		47H
FFFFF1D6H	Interrupt control register	UFIC1		√	√		47H
FFFFF1D8H	Interrupt control register	DMAIC6		√	√		47H
FFFFF1DAH	Interrupt control register	TB0OVBIC		√	√		47H
FFFFF1DCH	Interrupt control register	TB0CCBIC0		√	√		47H
FFFFF1DEH	Interrupt control register	TB1OVBIC		√	√		47H
FFFFF1E0H	Interrupt control register	TB1CCBIC0		√	√		47H
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H

(7/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D0 conversion result register 0	AD0CR0	R			√	0000H
FFFFF201H	A/D0 conversion result register 0H	AD0CR0H			√		00H
FFFFF202H	A/D0 conversion result register 1	AD0CR1				√	0000H
FFFFF203H	A/D0 conversion result register 1H	AD0CR1H			√		00H
FFFFF204H	A/D0 conversion result register 2	AD0CR2				√	0000H
FFFFF205H	A/D0 conversion result register 2H	AD0CR2H			√		00H
FFFFF206H	A/D0 conversion result register 3	AD0CR3				√	0000H
FFFFF207H	A/D0 conversion result register 3H	AD0CR3H			√		00H
FFFFF208H	A/D0 conversion result register 4	AD0CR4				√	0000H
FFFFF209H	A/D0 conversion result register 4H	AD0CR4H			√		00H
FFFFF20AH	A/D0 conversion result register 5	AD0CR5				√	0000H
FFFFF20BH	A/D0 conversion result register 5H	AD0CR5H			√		00H
FFFFF20CH	A/D0 conversion result register 6	AD0CR6				√	0000H
FFFFF20DH	A/D0 conversion result register 6H	AD0CR6H			√		00H
FFFFF20EH	A/D0 conversion result register 7	AD0CR7				√	0000H
FFFFF20FH	A/D0 conversion result register 7H	AD0CR7H			√		00H
FFFFF210H	A/D0 conversion result register 8	AD0CR8				√	0000H
FFFFF211H	A/D0 conversion result register 8H	AD0CR8H			√		00H
FFFFF212H	A/D0 conversion result register 9	AD0CR9				√	0000H
FFFFF213H	A/D0 conversion result register 9H	AD0CR9H			√		00H
FFFFF214H	A/D0 conversion result register 10	AD0CR10				√	0000H
FFFFF215H	A/D0 conversion result register 10H	AD0CR10H			√		00H
FFFFF216H	A/D0 conversion result register 11	AD0CR11				√	
FFFFF217H	A/D0 conversion result register 11H	AD0CR11H			√		00H
FFFFF218H	A/D0 conversion result register 12	AD0CR12				√	0000H
FFFFF219H	A/D0 conversion result register 12H	AD0CR12H			√		00H
FFFFF21AH	A/D0 conversion result register 13	AD0CR13				√	0000H
FFFFF21BH	A/D0 conversion result register 13H	AD0CR13H			√		00H
FFFFF21CH	A/D0 conversion result register 14	AD0CR14				√	0000H
FFFFF21DH	A/D0 conversion result register 14H	AD0CR14H			√		00H
FFFFF21EH	A/D0 conversion result register 15	AD0CR15				√	0000H
FFFFF21FH	A/D0 conversion result register 15H	AD0CR15H			√		00H
FFFFF220H	A/D converter 0 scan mode register	AD0SCM	R/W			√	0000H
FFFFF220H	A/D converter 0 scan mode register L	AD0SCML		√	√		00H
FFFFF221H	A/D converter 0 scan mode register H	AD0SCMH		√	√		00H
FFFFF222H	A/D converter 0 conversion time control register	AD0CTC		√	√		00H
FFFFF224H	A/D converter 0 conversion channel specification register	AD0CHEN				√	0000H
FFFFF224H	A/D converter 0 conversion channel specification register L	AD0CHENL		√	√		00H
FFFFF225H	A/D converter 0 conversion channel specification register H	AD0CHENH		√	√		00H

(8/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF230H	A/D converter 0 control register	AD0CTL0	R/W	√	√		00H
FFFFF231H	A/D converter 0 trigger select register	AD0TSEL		√	√		10H
FFFFF232H	A/D converter 0 channel specification register 1	AD0CH1		√	√		00H
FFFFF233H	A/D converter 0 channel specification register 2	AD0CH2		√	√		00H
FFFFF240H	A/D0 conversion result expansion register 0	AD0ECR0	R			√	0000H
FFFFF241H	A/D0 conversion result expansion register 0H	AD0ECR0H			√		00H
FFFFF242H	A/D0 conversion result expansion register 1	AD0ECR1				√	0000H
FFFFF243H	A/D0 conversion result expansion register 1H	AD0ECR1H			√		00H
FFFFF244H	A/D0 conversion result expansion register 2	AD0ECR2				√	0000H
FFFFF245H	A/D0 conversion result expansion register 2H	AD0ECR2H			√		00H
FFFFF246H	A/D0 conversion result expansion register 3	AD0ECR3				√	0000H
FFFFF247H	A/D0 conversion result expansion register 3H	AD0ECR3H			√		00H
FFFFF248H	A/D0 conversion result expansion register 4	AD0ECR4				√	0000H
FFFFF249H	A/D0 conversion result expansion register 4H	AD0ECR4H			√		00H
FFFFF254H	A/D converter 0 flag register	AD0FLG			√		00H
FFFFF255H	A/D converter 0 flag buffer register	AD0FLGB			√		00H
FFFFF260H	Operational amplifier 0 control register 0	OP0CTL0	R/W		√		00H
FFFFF261H	Comparator 0 control register 0	CMP0CTL0			√		00H
FFFFF262H	Comparator 0 control register 1	CMP0CTL1	R		√		00H
FFFFF263H	Comparator 0 control register 2	CMP0CTL2	R/W		√		00H
FFFFF264H	Comparator 0 control register 3	CMP0CTL3			√		00H
FFFFF270H	A/D converter 0 clock select register	AD0OCKS			√		00H
FFFFF274H	A/D converter 1 clock select register	AD1OCKS			√		00H
FFFFF278H	Comparator output digital noise elimination register 0L	CMPNFC0L			√		00H
FFFFF27AH	Comparator output digital noise elimination register 0F	CMPNFC0F			√		00H
FFFFF27CH	Comparator output digital noise elimination register 1L	CMPNFC1L			√		00H
FFFFF27EH	Comparator output digital noise elimination register 1F	CMPNFC1F			√		00H
FFFFF280H	A/D1 conversion result register 0	AD1CR0	R			√	0000H
FFFFF281H	A/D1 conversion result register 0H	AD1CR0H			√		00H
FFFFF282H	A/D1 conversion result register 1	AD1CR1				√	0000H
FFFFF283H	A/D1 conversion result register 1H	AD1CR1H			√		00H
FFFFF284H	A/D1 conversion result register 2	AD1CR2				√	0000H
FFFFF285H	A/D1 conversion result register 2H	AD1CR2H			√		00H
FFFFF286H	A/D1 conversion result register 3	AD1CR3				√	0000H
FFFFF287H	A/D1 conversion result register 3H	AD1CR3H			√		00H
FFFFF288H	A/D1 conversion result register 4	AD1CR4				√	0000H
FFFFF289H	A/D1 conversion result register 4H	AD1CR4H			√		00H
FFFFF28AH	A/D1 conversion result register 5	AD1CR5				√	0000H
FFFFF28BH	A/D1 conversion result register 5H	AD1CR5H			√		00H

(9/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF28CH	A/D1 conversion result register 6	AD1CR6	R			√	0000H
FFFFF28DH	A/D1 conversion result register 6H	AD1CR6H			√		00H
FFFFF28EH	A/D1 conversion result register 7	AD1CR7				√	0000H
FFFFF28FH	A/D1 conversion result register 7H	AD1CR7H			√		00H
FFFFF290H	A/D1 conversion result register 8	AD1CR8				√	0000H
FFFFF291H	A/D1 conversion result register 8H	AD1CR8H			√		00H
FFFFF292H	A/D1 conversion result register 9	AD1CR9				√	0000H
FFFFF293H	A/D1 conversion result register 9H	AD1CR9H			√		00H
FFFFF294H	A/D1 conversion result register 10	AD1CR10				√	0000H
FFFFF295H	A/D1 conversion result register 10H	AD1CR10H			√		00H
FFFFF296H	A/D1 conversion result register 11	AD1CR11				√	0000H
FFFFF297H	A/D1 conversion result register 11H	AD1CR11H			√		00H
FFFFF298H	A/D1 conversion result register 12	AD1CR12				√	0000H
FFFFF299H	A/D1 conversion result register 12H	AD1CR12H			√		00H
FFFFF29AH	A/D1 conversion result register 13	AD1CR13				√	0000H
FFFFF29BH	A/D1 conversion result register 13H	AD1CR13H			√		00H
FFFFF29CH	A/D1 conversion result register 14	AD1CR14				√	0000H
FFFFF29DH	A/D1 conversion result register 14H	AD1CR14H			√		00H
FFFFF29EH	A/D1 conversion result register 15	AD1CR15				√	0000H
FFFFF29FH	A/D1 conversion result register 15H	AD1CR15H			√		00H
FFFFF2A0H	A/D converter 1 scan mode register	AD1SCM	R/W			√	0000H
FFFFF2A0H	A/D converter 1 scan mode register L	AD1SCML		√	√		00H
FFFFF2A1H	A/D converter 1 scan mode register H	AD1SCMH		√	√		00H
FFFFF2A2H	A/D converter 1 conversion time control register	AD1CTC		√	√		00H
FFFFF2A4H	A/D converter 1 conversion channel specification register	AD1CHEN				√	0000H
FFFFF2A4H	A/D converter 1 conversion channel specification register L	AD1CHENL		√	√		00H
FFFFF2A5H	A/D converter 1 conversion channel specification register H	AD1CHENH		√	√		00H
FFFFF2B0H	A/D converter 1 control register	AD1CTL0		√	√		00H
FFFFF2B1H	A/D converter 1 trigger select register	AD1TSEL		√	√		10H
FFFFF2B2H	A/D converter 1 channel specification register 1	AD1CH1		√	√		00H
FFFFF2B3H	A/D converter 1 channel specification register 2	AD1CH2		√	√		00H
FFFFF2C0H	A/D1 conversion result expansion register 0	AD1ECR0	R			√	0000H
FFFFF2C1H	A/D1 conversion result expansion register 0H	AD1ECR0H			√		00H
FFFFF2C2H	A/D1 conversion result expansion register 1	AD1ECR1				√	0000H
FFFFF2C3H	A/D1 conversion result expansion register 1H	AD1ECR1H			√		00H
FFFFF2C4H	A/D1 conversion result expansion register 2	AD1ECR2				√	0000H
FFFFF2C5H	A/D1 conversion result expansion register 2H	AD1ECR2H			√		00H
FFFFF2C6H	A/D1 conversion result expansion register 3	AD1ECR3				√	0000H
FFFFF2C7H	A/D1 conversion result expansion register 3H	AD1ECR3H			√		00H

(10/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF2C8H	A/D1 conversion result expansion register 4	AD1ECR4	R			√	0000H
FFFFF2C9H	A/D1 conversion result expansion register 4H	AD1ECR4H			√		00H
FFFFF2D4H	A/D converter 1 flag register	AD1FLG			√		00H
FFFFF2D5H	A/D converter 1 flag buffer register	AD1FLGB			√		00H
FFFFF2E0H	Operational amplifier 1 control register 0	OP1CTL0	R/W		√		00H
FFFFF2E1H	Comparator 1 control register 0	CMP1CTL0			√		00H
FFFFF2E2H	Comparator 1 control register 1	CMP1CTL1	R		√		00H
FFFFF2E3H	Comparator 1 control register 2	CMP1CTL2	R/W		√		00H
FFFFF2E4H	Comparator 1 control register 3	CMP1CTL3			√		00H
FFFFF2F0H	A/D trigger falling edge specification register	ADTF		√	√		00H
FFFFF2F2H	A/D trigger rising edge specification register	ADTR		√	√		00H
FFFFF2F4H	Comparator output interrupt falling edge specification register	CMPOF		√	√		00H
FFFFF2F6H	Comparator output interrupt rising edge specification register	CMPOR		√	√		00H
FFFFF2F8H	A/DLDTRG1 input select register	ADLTS1			√		00H
FFFFF2FAH	A/DLDTRG2 input select register	ADLTS2			√		00H
FFFFF310H	Digital noise elimination 0 control register 00	INTNFC00			√		00H
FFFFF312H	Digital noise elimination 0 control register 01	INTNFC01			√		00H
FFFFF314H	Digital noise elimination 0 control register 02	INTNFC02			√		00H
FFFFF318H	Digital noise elimination 0 control register 17	INTNFC17			√		00H
FFFFF31AH	Digital noise elimination 0 control register 18	INTNFC18			√		00H
FFFFF31CH	Digital noise elimination 0 control register 19	INTNFC19			√		00H
FFFFF340H	DMA wait control register 0	DMAWC0		√	√		37H
FFFFF342H	DMA wait control register 1	DMAWC1		√	√		07H
FFFFF3A0H	Port DL function control register	PFCDL				√	0000H
FFFFF3A0H	Port DL function control register L	PFCDLL		√	√		00H
FFFFF3A1H	Port DL function control register H	PFCDLH		√	√		00H
FFFFF3C0H	Port DL function control expansion register	PFCEDL				√	0000H
FFFFF3C0H	Port DL function control expansion register L	PFCEDLL		√	√		00H
FFFFF3C1H	Port DL function control expansion register H	PFCEDLH		√	√		00H
FFFFF400H	Port 0 register	P0		√	√		Undefined
FFFFF402H	Port 1 register	P1		√	√		Undefined
FFFFF404H	Port 2 register	P2		√	√		Undefined
FFFFF406H	Port 3 register	P3		√	√		Undefined
FFFFF408H	Port 4 register	P4		√	√		Undefined
FFFFF40AH	Port 5 register	P5		√	√		Undefined
FFFFF412H	Port 9 register	P9[Note]		√	√		Undefined
FFFFF420H	Port 0 mode register	PM0		√	√		FFH
FFFFF422H	Port 1 mode register	PM1		√	√		FFH
FFFFF424H	Port 2 mode register	PM2		√	√		FFH
FFFFF426H	Port 3 mode register	PM3		√	√		FFH

(11/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF428H	Port 4 mode register	PM4	R/W	√	√		FFH
FFFFF42AH	Port 5 mode register	PM5		√	√		FFH
FFFFF432H	Port 9 mode register	PM9[Note]		√	√		FFH
FFFFF440H	Port 0 mode control register	PMC0		√	√		00H
FFFFF442H	Port 1 mode control register	PMC1		√	√		00H
FFFFF444H	Port 2 mode control register	PMC2		√	√		00H
FFFFF446H	Port 3 mode control register	PMC3		√	√		00H
FFFFF448H	Port 4 mode control register	PMC4		√	√		00H
FFFFF44AH	Port 5 mode control register	PMC5		√	√		00H
FFFFF452H	Port 9 mode control register	PMC9[Note]		√	√		00H
FFFFF460H	Port 0 function control register	PFC0		√	√		00H
FFFFF462H	Port 1 function control register	PFC1		√	√		00H
FFFFF464H	Port 2 function control register	PFC2		√	√		00H
FFFFF466H	Port 3 function control register	PFC3		√	√		00H
FFFFF468H	Port 4 function control register	PFC4		√	√		00H
FFFFF46AH	Port 5 function control register	PFC5		√	√		00H
FFFFF480H	Bus cycle type configuration register 0	BCT0				√	CCCCH
FFFFF484H	Data wait control register 0	DWC0				√	7777H
FFFFF488H	Address wait control register	AWC				√	FFFFH
FFFFF48AH	Bus cycle control register	BCC				√	AAAAH
FFFFF48EH	Bus clock division control register	DVC			√		83H
FFFFF540H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF544H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF550H	TMM1 control register 0	TM1CTL0		√	√		00H
FFFFF554H	TMM1 compare register 0	TM1CMP0				√	0000H
FFFFF560H	TMM2 control register 0	TM2CTL0		√	√		00H
FFFFF564H	TMM2 compare register 0	TM2CMP0				√	0000H
FFFFF570H	TMM3 control register 0	TM3CTL0		√	√		00H
FFFFF574H	TMM3 compare register 0	TM3CMP0				√	0000H
FFFFF580H	TMT0 control register 0	TT0CTL0		√	√		00H
FFFFF581H	TMT0 control register 1	TT0CTL1		√	√		00H
FFFFF582H	TMT0 control register 2	TT0CTL2		√	√		00H
FFFFF583H	TMT0 I/O control register 0	TT0IOC0		√	√		00H
FFFFF584H	TMT0 I/O control register 1	TT0IOC1		√	√		00H
FFFFF585H	TMT0 I/O control register 2	TT0IOC2		√	√		00H
FFFFF586H	TMT0 I/O control register 3	TT0IOC3		√	√		00H
FFFFF587H	TMT0 option register 0	TT0OPT0		√	√		00H
FFFFF588H	TMT0 option register 1	TT0OPT1		√	√		00H
FFFFF58AH	TMT0 capture/compare register 0	TT0CCR0				√	0000H
FFFFF58CH	TMT0 capture/compare register 1	TT0CCR1				√	0000H

Note V850E/IH4-H only

(12/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF58EH	TMT0 counter read buffer register	TT0CNT	R			√	0000H
FFFFF590H	TMT0 counter write register	TT0TCW	R/W			√	0000H
FFFFF5A0H	Digital noise elimination 2 control register 0	TTNFC0			√		00H
FFFFF5A2H	Digital noise elimination 2 control register 1	TTNFC1			√		00H
FFFFF5A4H	TMT0 capture input select register	TTISL0			√		Undefined
FFFFF5A6H	TMT1 capture input select register	TTISL1			√		Undefined
FFFFF5C0H	TMT1 control register 0	TT1CTL0		√	√		00H
FFFFF5C1H	TMT1 control register 1	TT1CTL1		√	√		00H
FFFFF5C2H	TMT1 control register 2	TT1CTL2		√	√		00H
FFFFF5C3H	TMT1 I/O control register 0	TT1IOC0		√	√		00H
FFFFF5C4H	TMT1 I/O control register 1	TT1IOC1		√	√		00H
FFFFF5C5H	TMT1 I/O control register 2	TT1IOC2		√	√		00H
FFFFF5C6H	TMT1 I/O control register 3	TT1IOC3		√	√		00H
FFFFF5C7H	TMT1 option register 0	TT1OPT0		√	√		00H
FFFFF5C8H	TMT1 option register 1	TT1OPT1		√	√		00H
FFFFF5CAH	TMT1 capture/compare register 0	TT1CCR0				√	0000H
FFFFF5CCH	TMT1 capture/compare register 1	TT1CCR1				√	0000H
FFFFF5CEH	TMT1 counter read buffer register	TT1CNT	R			√	0000H
FFFFF5D0H	TMT1 counter write register	TT1TCW	R/W			√	0000H
FFFFF5E0H	TAB0 control register 0	TAB0CTL0		√	√		00H
FFFFF5E1H	TAB0 control register 1	TAB0CTL1		√	√		00H
FFFFF5E2H	TAB0 I/O control register 0	TAB0IOC0		√	√		00H
FFFFF5E3H	TAB0 I/O control register 1	TAB0IOC1		√	√		00H
FFFFF5E4H	TAB0 I/O control register 2	TAB0IOC2		√	√		00H
FFFFF5E5H	TAB0 option register 0	TAB0OPT0		√	√		00H
FFFFF5E6H	TAB0 capture/compare register 0	TAB0CCR0				√	0000H
FFFFF5E8H	TAB0 capture/compare register 1	TAB0CCR1				√	0000H
FFFFF5EAH	TAB0 capture/compare register 2	TAB0CCR2				√	0000H
FFFFF5ECH	TAB0 capture/compare register 3	TAB0CCR3				√	0000H
FFFFF5EEH	TAB0 counter read buffer register	TAB0CNT	R			√	0000H
FFFFF600H	TAB0 option register 1	TAB0OPT1	R/W	√	√		00H
FFFFF601H	TAB0 option register 2	TAB0OPT2		√	√		00H
FFFFF602H	TAB0 I/O control register 3	TAB0IOC3		√	√		A8H
FFFFF603H	TAB0 option register 3	TAB0OPT3		√	√		00H
FFFFF604H	TAB0 deadtime compare register	TAB0DTC				√	0000H
FFFFF610H	High-impedance output control register 00	HZA0CTL0		√	√		00H
FFFFF611H	High-impedance output control register 01	HZA0CTL1		√	√		00H
FFFFF618H	High-impedance output control register 10	HZA1CTL0[Note]		√	√		00H
FFFFF619H	High-impedance output control register 11	HZA1CTL1[Note]		√	√		00H
FFFFF620H	TAB1 control register 0	TAB1CTL0		√	√		00H

Note V850E/IH4-H only

(13/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF621H	TAB1 control register 1	TAB1CTL1	R/W	√	√		00H
FFFFF622H	TAB1 I/O control register 0	TAB1IOC0		√	√		00H
FFFFF623H	TAB1 I/O control register 1	TAB1IOC1[Note]		√	√		00H
FFFFF624H	TAB1 I/O control register 2	TAB1IOC2		√	√		00H
FFFFF625H	TAB1 option register 0	TAB1OPT0		√	√		00H
FFFFF626H	TAB1 capture/compare register 0	TAB1CCR0				√	0000H
FFFFF628H	TAB1 capture/compare register 1	TAB1CCR1				√	0000H
FFFFF62AH	TAB1 capture/compare register 2	TAB1CCR2				√	0000H
FFFFF62CH	TAB1 capture/compare register 3	TAB1CCR3				√	0000H
FFFFF62EH	TAB1 counter read buffer register	TAB1CNT	R			√	0000H
FFFFF640H	TAB1 option register 1	TAB1OPT1[Note]	R/W	√	√		00H
FFFFF641H	TAB1 option register 2	TAB1OPT2[Note]		√	√		00H
FFFFF642H	TAB1 I/O control register 3	TAB1IOC3[Note]		√	√		A8H
FFFFF643H	TAB1 option register 3	TAB1OPT3[Note]		√	√		00H
FFFFF644H	TAB1 deadtime compare register	TAB1DTC[Note]				√	0000H
FFFFF650H	High-impedance output control register 20	HZA2CTL0		√	√		00H
FFFFF651H	High-impedance output control register 21	HZA2CTL1		√	√		00H
FFFFF658H	High-impedance output control register 30	HZA3CTL0[Note]		√	√		00H
FFFFF659H	High-impedance output control register 31	HZA3CTL1[Note]		√	√		00H
FFFFF660H	TAA0 control register 0	TAA0CTL0		√	√		00H
FFFFF661H	TAA0 control register 1	TAA0CTL1		√	√		00H
FFFFF662H	TAA0 I/O control register 0	TAA0IOC0		√	√		00H
FFFFF665H	TAA0 option register 0	TAA0OPT0		√	√		00H
FFFFF666H	TAA0 capture/compare register 0	TAA0CCR0				√	0000H
FFFFF668H	TAA0 capture/compare register 1	TAA0CCR1				√	0000H
FFFFF66AH	TAA0 counter read buffer register	TAA0CNT	R			√	0000H
FFFFF680H	TAA1 control register 0	TAA1CTL0	R/W	√	√		00H
FFFFF681H	TAA1 control register 1	TAA1CTL1		√	√		00H
FFFFF682H	TAA1 I/O control register 0	TAA1IOC0		√	√		00H
FFFFF685H	TAA1 option register 0	TAA1OPT0		√	√		00H
FFFFF686H	TAA1 capture/compare register 0	TAA1CCR0				√	0000H
FFFFF688H	TAA1 capture/compare register 1	TAA1CCR1				√	0000H
FFFFF68AH	TAA1 counter read buffer register	TAA1CNT	R			√	0000H
FFFFF6A0H	TAA2 control register 0	TAA2CTL0	R/W	√	√		00H
FFFFF6A1H	TAA2 control register 1	TAA2CTL1		√	√		00H
FFFFF6A2H	TAA2 I/O control register 0	TAA2IOC0		√	√		00H
FFFFF6A3H	TAA2 I/O control register 1	TAA2IOC1		√	√		00H
FFFFF6A4H	TAA2 I/O control register 2	TAA2IOC2		√	√		00H
FFFFF6A5H	TAA2 option register 0	TAA2OPT0		√	√		00H
FFFFF6A6H	TAA2 capture/compare register 0	TAA2CCR0				√	0000H

Note V850E/IH4-H only

(14/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF6A8H	TAA2 capture/compare register 1	TAA2CCR1	R/W			√	0000H
FFFFF6AAH	TAA2 counter read buffer register	TAA2CNT	R			√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS	R/W		√		05H
FFFFF6D0H	Watchdog timer mode register	WDTM			√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		1AH
FFFFF700H	Port 0 function control expansion register	PFCE0		√	√		00H
FFFFF702H	Port 1 function control expansion register	PFCE1		√	√		00H
FFFFF704H	Port 2 function control expansion register	PFCE2		√	√		00H
FFFFF706H	Port 3 function control expansion register	PFCE3		√	√		00H
FFFFF708H	Port 4 function control expansion register	PFCE4		√	√		00H
FFFFF70AH	Port 5 function control expansion register	PFCE5		√	√		00H
FFFFF780H	TMT2 control register 0	TT2CTL0	R/W	√	√		00H
FFFFF781H	TMT2 control register 1	TT2CTL1		√	√		00H
FFFFF783H	TMT2 I/O control register 0	TT2IOC0		√	√		00H
FFFFF784H	TMT2 I/O control register 1	TT2IOC1		√	√		00H
FFFFF785H	TMT2 I/O control register 2	TT2IOC2		√	√		00H
FFFFF787H	TMT2 option register 0	TT2OPT0		√	√		00H
FFFFF78AH	TMT2 capture/compare register 0	TT2CCR0				√	0000H
FFFFF78CH	TMT2 capture/compare register 1	TT2CCR1				√	0000H
FFFFF78EH	TAA2 counter read buffer register	TT2CNT	R			√	0000H
FFFFF7A0H	Digital noise elimination 3 control register 2	TTNFC2	R/W		√		00H
FFFFF7A2H	Digital noise elimination 3 control register 3	TTNFC3			√		00H
FFFFF7C0H	TMT3 control register 0	TT3CTL0		√	√		00H
FFFFF7C1H	TMT3 control register 1	TT3CTL1		√	√		00H
FFFFF7C3H	TMT3 I/O control register 0	TT3IOC0		√	√		00H
FFFFF7C4H	TMT3 I/O control register 1	TT3IOC1		√	√		00H
FFFFF7C5H	TMT3 I/O control register 2	TT3IOC2		√	√		00H
FFFFF7C7H	TMT3 option register 0	TT3OPT0		√	√		00H
FFFFF7CAH	TMT3 capture/compare register 0	TT3CCR0				√	0000H
FFFFF7CCH	TMT3 capture/compare register 1	TT3CCR1				√	0000H
FFFFF7CEH	TMT3 counter read buffer register	TT3CNT	R			√	0000H
FFFFF802H	System status register	SYS	R/W	√	√		00H
FFFFF820H	Power save mode register	PSMR		√	√		00H
FFFFF828H	Processor clock control register	PCC		√	√		03H
FFFFF82CH	PLL control register	PLLCTL		√	√		01H
FFFFF870H	Clock monitor mode register	CLM		√	√		00H
FFFFF888H	Reset source flag register	RESF		√	√		00H/10H/01H
FFFFF890H	Low-voltage detection register	LVIM		√	√		00H
FFFFF891H	Low-voltage detection level select register	LVIS			√		00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1			√		00H

(15/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFFA02H	UARTA0 control register 2	UA0CTL2	R/W		√		FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 status register	UA0STR		√	√		00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√		FFH
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2			√		FFH
FFFFFA13H	UARTA1 option control register 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 status register	UA1STR		√	√		00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√		FFH
FFFFFA17H	UARTA1 transmit data register	UA1TX	R/W		√		FFH
FFFFFA20H	UARTA2 control register 0	UA2CTL0		√	√		10H
FFFFFA21H	UARTA2 control register 1	UA2CTL1			√		00H
FFFFFA22H	UARTA2 control register 2	UA2CTL2			√		FFH
FFFFFA23H	UARTA2 option control register 0	UA2OPT0		√	√		14H
FFFFFA24H	UARTA2 status register	UA2STR		√	√		00H
FFFFFA26H	UARTA2 receive data register	UA2RX	R		√		FFH
FFFFFA27H	UARTA2 transmit data register	UA2TX	R/W		√		FFH
FFFFFA40H	UARTB control register 0	UBCTL0		√	√		10H
FFFFFA42H	UARTB control register 2	UBCTL2				√	FFFFH
FFFFFA44H	UARTB status register	UBSTR		√	√		00H
FFFFFA46H	UARTB receive data register AP	UBRXAP	R			√	00FFH
FFFFFA46H	UARTB receive data register	UBRX			√		FFH
FFFFFA48H	UARTB transmit data register	UBTX	W		√		FFH
FFFFFA4AH	UARTBFIFO control register 0	UBFIC0	R/W	√	√		00H
FFFFFA4BH	UARTBFIFO control register 1	UBFIC1		√	√		00H
FFFFFA4CH	UARTBFIFO control register 2	UBFIC2				√	0000H
FFFFFA4CH	UARTBFIFO control register 2L	UBFIC2L			√		00H
FFFFFA4DH	UARTBFIFO control register 2H	UBFIC2H			√		00H
FFFFFA4EH	UARTBFIFO status register 0	UBFIS0	R		√		00H
FFFFFA4FH	UARTBFIFO status register 1	UBFIS1			√		10H
FFFFFB00H	D/A converter 0 conversion value setting register 0	DA0CS0	R/W		√		00H
FFFFFB01H	D/A converter 0 conversion value setting register 1	DA0CS1			√		00H
FFFFFB02H	D/A converter 0 mode register	DA0M		√	√		00H
FFFFFB10H	D/A converter 1 conversion value setting register 0	DA1CS0			√		00H
FFFFFB11H	D/A converter 1 conversion value setting register 1	DA1CS1			√		00H
FFFFFB12H	D/A converter 1 mode register	DA1M		√	√		00H
FFFFFB40H	Digital noise elimination 1 control register 2	TANFC2			√		00H
FFFFFB80H	A/D converter 2 mode register 0	AD2M0		√	√		00H
FFFFFB81H	A/D converter 2 mode register 1	AD2M1		√	√		00H
FFFFFB82H	A/D converter 2 channel specification register	AD2S		√	√		00H

(16/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFFB90H	A/D2 conversion result register 0	AD2CR0	R			√	0000H
FFFFFB91H	A/D2 conversion result register 0H	AD2CR0H			√		00H
FFFFFB92H	A/D2 conversion result register 1	AD2CR1				√	0000H
FFFFFB93H	A/D2 conversion result register 1H	AD2CR1H			√		00H
FFFFFB94H	A/D2 conversion result register 2	AD2CR2				√	0000H
FFFFFB95H	A/D2 conversion result register 2H	AD2CR2H			√		00H
FFFFFB96H	A/D2 conversion result register 3	AD2CR3				√	0000H
FFFFFB97H	A/D2 conversion result register 3H	AD2CR3H			√		00H
FFFFFB98H	A/D2 conversion result register 4	AD2CR4				√	0000H
FFFFFB99H	A/D2 conversion result register 4H	AD2CR4H			√		00H
FFFFFB9AH	A/D2 conversion result register 5	AD2CR5				√	0000H
FFFFFB9BH	A/D2 conversion result register 5H	AD2CR5H			√		00H
FFFFFB9CH	A/D2 conversion result register 6	AD2CR6				√	0000H
FFFFFB9DH	A/D2 conversion result register 6H	AD2CR6H			√		00H
FFFFFB9EH	A/D2 conversion result register 7	AD2CR7				√	0000H
FFFFFB9FH	A/D2 conversion result register 7H	AD2CR7H			√		00H
FFFFFBA0H	A/D2 conversion result register 8	AD2CR8				√	0000H
FFFFFBA1H	A/D2 conversion result register 8H	AD2CR8H			√		00H
FFFFFBA2H	A/D2 conversion result register 9	AD2CR9				√	0000H
FFFFFBA3H	A/D2 conversion result register 9H	AD2CR9H			√		00H
FFFFFBA4H	A/D2 conversion result register 10	AD2CR10				√	0000H
FFFFFBA5H	A/D2 conversion result register 10H	AD2CR10H			√		00H
FFFFFBA6H	A/D2 conversion result register 10	AD2CR10				√	0000H
FFFFFBA7H	A/D2 conversion result register 11H	AD2CR11H			√		00H
FFFFFBB0H	Port 7 register L	P7L		√	√		Undefined
FFFFFBB1H	Port 7 register H	P7H		√	√		Undefined
FFFFFBB8H	Port 7 mode control register L	PMC7L	R/W	√	√		00H
FFFFFBB9H	Port 7 mode control register H	PMC7H		√	√		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H
FFFFFC02H	External interrupt falling edge specification register 1	INTF1		√	√		00H
FFFFFC04H	External interrupt falling edge specification register 2	INTF2		√	√		00H
FFFFFC06H	External interrupt falling edge specification register 3	INTF3		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H
FFFFFC22H	External interrupt rising edge specification register 1	INTR1		√	√		00H
FFFFFC24H	External interrupt rising edge specification register 2	INTR2		√	√		00H
FFFFFC26H	External interrupt rising edge specification register 3	INTR3		√	√		00H
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		00H
FFFFFC42H	Pull-up resistor option register 1	PU1		√	√		00H
FFFFFC44H	Pull-up resistor option register 2	PU2		√	√		00H
FFFFFC46H	Pull-up resistor option register 3	PU3		√	√		00H
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		00H

(17/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFFC4AH	Pull-up resistor option register 5	PU5	R/W	√	√		00H
FFFFFC52H	Pull-up resistor option register 9	PU9[Note]		√	√		00H
FFFFFC66H	Port 3 function register	PF3		√	√		00H
FFFFFD00H	CSIF0 control register 0	CF0CTL0		√	√		01H
FFFFFD01H	CSIF0 control register 1	CF0CTL1		√	√		00H
FFFFFD02H	CSIF0 control register 2	CF0CTL2			√		00H
FFFFFD03H	CSIF0 status register	CF0STR		√	√		00H
FFFFFD04H	CSIF0 receive data register	CF0RX	R			√	0000H
FFFFFD04H	CSIF0 receive data register L	CF0RXL			√		00H
FFFFFD06H	CSIF0 transmit data register	CF0TX	R/W			√	0000H
FFFFFD06H	CSIF0 transmit data register L	CF0TXL			√		00H
FFFFFD10H	CSIF1 control register 0	CF1CTL0		√	√		01H
FFFFFD11H	CSIF1 control register 1	CF1CTL1		√	√		00H
FFFFFD12H	CSIF1 control register 2	CF1CTL2			√		00H
FFFFFD13H	CSIF1 status register	CF1STR		√	√		00H
FFFFFD14H	CSIF1 receive data register	CF1RX	R			√	0000H
FFFFFD14H	CSIF1 receive data register L	CF1RXL			√		00H
FFFFFD16H	CSIF1 transmit data register	CF1TX	R/W			√	0000H
FFFFFD16H	CSIF1 transmit data register L	CF1TXL			√		00H
FFFFFD20H	CSIF2 control register 0	CF2CTL0		√	√		01H
FFFFFD21H	CSIF2 control register 1	CF2CTL1		√	√		00H
FFFFFD22H	CSIF2 control register 2	CF2CTL2			√		00H
FFFFFD23H	CSIF2 status register	CF2STR		√	√		00H
FFFFFD24H	CSIF2 receive data register	CF2RX	R			√	0000H
FFFFFD24H	CSIF2 receive data register L	CF2RXL			√		00H
FFFFFD26H	CSIF2 transmit data register	CF2TX	R/W			√	0000H
FFFFFD26H	CSIF2 transmit data register L	CF2TXL			√		00H
FFFFFD80H	IIC shift register 0	IIC0			√		00H
FFFFFD82H	IIC control register 0	IICC0		√	√		00H
FFFFFD83H	Slave address register 0	SVA0			√		00H
FFFFFD84H	IIC clock select register 0	IICCL0		√	√		00H
FFFFFD85H	IIC function expansion register 0	IICX0		√	√		00H
FFFFFD86H	IIC status register 0	IICS0	R	√	√		00H
FFFFFD8AH	IIC flag register 0	IICF0	R/W	√	√		00H
FFFFFD90H	IIC OPS clock select register	IICOCKS			√		00H
FFFFFE00H	High-impedance output control register 40	HZA4CTL0		√	√		00H
FFFFFE01H	High-impedance output control register 41	HZA4CTL1[Note]		√	√		00H
FFFFFE08H	High-impedance output control register 50	HZA5CTL0		√	√		00H
FFFFFE09H	High-impedance output control register 51	HZA5CTL1[Note]		√	√		00H
FFFFFE10H	High-impedance output control register 60	HZA6CTL0		√	√		00H

Note V850E/IH4-H only

(18/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFFE11H	High-impedance output control register 61	HZA6CTL1	R/W	√	√		00H
FFFFFE18H	High-impedance output control register 70	HZA7CTL0[Note]		√	√		00H
FFFFFE19H	High-impedance output control register 71	HZA7CTL1[Note]		√	√		00H
FFFFFE20H	High-impedance output control register 80	HZA8CTL0		√	√		00H
FFFFFE21H	High-impedance output control register 81	HZA8CTL1[Note]		√	√		00H
FFFFFE28H	High-impedance output control register 90	HZA9CTL0		√	√		00H
FFFFFE29H	High-impedance output control register 91	HZA9CTL1[Note]		√	√		00H
FFFFFE30H	High-impedance output control register 100	HZA10CTL0		√	√		00H
FFFFFE31H	High-impedance output control register 101	HZA10CTL1		√	√		00H
FFFFFE38H	High-impedance output control register 110	HZA11CTL0[Note]		√	√		00H
FFFFFE39H	High-impedance output control register 111	HZA11CTL1[Note]		√	√		00H
FFFFFE40H	High-impedance output control register 120	HZA12CTL0		√	√		00H
FFFFFE41H	High-impedance output control register 121	HZA12CTL1[Note]		√	√		00H
FFFFFF44H	Pull-up resistor option register DL	PUDL				√	0000H
FFFFFF44H	Pull-up resistor option register DLL	PUDLL		√	√		00H
FFFFFF45H	Pull-up resistor option register DLH	PUDLH		√	√		00H
FFFFFE80H	USB clock selection register	UCKSEL		√	√		00H
FFFFFE81H	USB function control register	UFCTL		√	√		03H

Note V850E/IH4-H only

3.4.8 Special registers

Special registers are registers that are protected from being written with illegal data due to a program loop.

The V850E/IG4-H and V850E/IH4-H have the following five special registers.

- Power save control register (PSC)
- Processor clock control register (PCC)
- Reset source flag register (RESF)
- Clock monitor mode register (CLM)
- Low-voltage detection register (LVIM)

In addition, a command register (PRCMD) is provided to protect against a write access to the special registers so that the application system does not inadvertently stop due to a program loop. A write access to the special registers is made in a specific sequence, and an illegal store operation is reported to the system status register (SYS).

(1) Setting data to special registers

Set data to the special registers in the following sequence.

<1> Prepare data to be set to the special register in a general-purpose register.
<2> Write the data prepared in <1> to the command register.
<3> Write the setting data to the special register (by using the following instructions).

- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

(<4> to <8> Insert NOP instructions (5 instructions).)[Note]

[Example] With PSC register (setting standby mode)

```
   ST.B r11, PSMR[r0]   ; Set PSMR register (setting IDLE and STOP modes).
<1>MOV 0x02, r10
<2>ST.B r10, PRCMD[r0]  ; Write PRCMD register.
<3>ST.B r10, PSC[r0]    ; Set PSC register.
<4>NOP[Note]            ; Dummy instruction
<5>NOP[Note]            ; Dummy instruction
<6>NOP[Note]            ; Dummy instruction
<7>NOP[Note]            ; Dummy instruction
<8>NOP[Note]            ; Dummy instruction
  (next instruction)
```

There is no special sequence to read a special register.

Note Five NOP instructions or more must be inserted immediately after setting the IDLE mode or STOP mode (by setting the PSC.STB bit to 1).

Cautions 1. When a store instruction is executed to store data in the command register, interrupts are not acknowledged. This is because it is assumed that steps <2> and <3> above are performed by successive store instructions. If another instruction is placed between <2> and <3>, and if an interrupt is acknowledged by that instruction, the above sequence may not be established, causing malfunction.

2. Although dummy data is written to the command register, use the same general-purpose register used to set the special register (<3> in Example) by using the store instruction to write data to the command register (<2> in Example). The same applies when a general-purpose register is used for addressing.

An example of setting the special register (<3> in Example) by using the bit manipulation instruction is shown below.

```
CLR1 4, RESF[r0]
```

3. Before executing this processing, terminate all DMA transfer operations.

(2) Command register (PRCMD)

The PRCMD register is an 8-bit register that protects the registers that may seriously affect the application system from being written, so that the system does not inadvertently stop due to a program loop. The first write access to a special register is valid after data has been written in advance to the PRCMD register. In this way, the value of the special register can be rewritten only in a specific sequence, so as to protect the register from an illegal write access.

An illegal write operation to a special register can be checked by using the SYS.PRERR bit.

The PRCMD register is write-only, in 8-bit units (undefined data is read when this register is read).

Reset makes this register undefined.

After reset: Undefined W Address: FFFFF1FCH

	7	6	5	4	3	2	1	0
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) System status register (SYS)

Status flags that indicate the operation status of the overall system are allocated to this register.

If this register is not written in the correct sequence including an access to the PRCMD register, data is not written to the intended register, a protection error occurs, and the PRERR flag is set. This register is cleared by writing “0” to it by an instruction from CPU.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF802H

	7	6	5	4	3	2	1	<0>
SYS	0	0	0	0	0	0	0	PRERR

PRERR	Protection error detection
0	Protection error did not occur.
1	Protection error occurred.

The PRERR flag operates under the following conditions.

(a) Set condition (PRERR flag = 1)

- When data is written to a special register without writing anything to the PRCMD register (when <3> is executed without executing <2> in **3.4.8 (1) Setting data to special registers**)
- When data is written to an on-chip peripheral I/O register other than a special register (including execution of a bit manipulation instruction) after writing data to the PRCMD register (if <3> in **3.4.8 (1) Setting data to special registers** is not the setting of a special register)

Remark Even if an on-chip peripheral I/O register is read (excluding execution of a bit manipulation instruction) between a write access to the PRCMD register and a write access to a special register (such as an access to the internal RAM), the PRERR flag is not set and data can be written to the special register.

(b) Clear condition (PRERR flag = 0)

(i) When 0 is written to the SYS.PRERR flag

(ii) When the system is reset

Cautions 1. If 0 is written to the SYS.PRERR bit which is not a special register, immediately after a write access to the PRCMD register, the PRERR bit is cleared to 0 (the write access takes precedence).

2. If data is written to the PRCMD register, which is not a special register, immediately after a write access to the PRCMD register, the PRERR bit is set to 1.

3.4.9 System wait control register (VSWC)

The VSWC register is a register that controls the bus access wait for the on-chip peripheral I/O registers.

Access to on-chip peripheral I/O registers of the V850E1 CPU core is basically made in 3 clocks; however, in the V850E/IG4-H and V850E/IH4-H, a wait set by the VSWC register is required in addition to those 3 clocks. Set 12H (set wait for 3 clocks) to VSWC.

This register can be read or written in 8-bit units (address: FFFFF06EH, initial value: 77H).

CPU Clock Frequency (f_{CPU})	VSWC Set Value
1.25 MHz ≤ f_{CPU} ≤ 100 MHz	12H

Caution When using the V850E/IG4-H or V850E/IH4-H, the VSWC register must be set first. Set other registers if necessary after setting the VSWC register.

Remark When a register includes status flags that indicate the statuses of the on-chip peripheral functions (such as UAnSTR) or a register that indicates the count value of a timer (such as TAAnCNT) is accessed, a register access retry operation takes place if the timing at which the flag and count value changes and the timing of the register access overlap. Consequently, access to the on-chip peripheral I/O register may take a long time.

3.4.10 DMA wait control registers 0, 1 (DMAWC0, DMAWC1)

Set the DMAWCn registers to the following values:

DMAWC0 register value: 12H
DMAWC1 register value: 00H

This registers can be read or written in 8-bit units.

Register Name	Address	Initial Value
DMAWC0 register	FFFFF340H	37H
DMAWC1 register	FFFFF342H	07H

CHAPTER 4 PORT FUNCTIONS

4.1 Features

4.1.1 V850E/IG4-H

○ Input-only ports: 12
I/O ports: 51

○ Input data read/output data write is enabled in 1-bit units.

○ On-chip pull-up resistor can be connected in 1-bit units (ports 0 to 5 and DL only).
However, an on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode. An on-chip pull-up resistor can also be connected to the TOT21, TOT31, TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, TOB1T3, and TOB1B3 pins, which function as output pins in the alternate-function mode, when these pins go into a high-impedance state due to a signal input to the TOT2OFF, TOT3OFF, TOB0OFF, TOB1OFF, or TOB01OFF pin or software processing.

4.1.2 V850E/IH4-H

○ Input-only ports: 12
I/O ports: 68

○ Input data read/output data write is enabled in 1-bit units.

○ On-chip pull-up resistor can be connected in 1-bit units (ports 0 to 5, 9 and DL only).
However, an on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode. An on-chip pull-up resistor can also be connected to the TOT21, TOT31, TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, TOB1T1 to TOB1T3, and TOB1B1 to TOB1B3 pins, which function as output pins in the alternate-function mode, when these pins go into a high-impedance state due to a signal input to the TOT2OFF, TOT3OFF, TOB0OFF, TOB1OFF, or TOB01OFF pin or software processing.

4.2 Port Configuration

4.2.1 V850E/IG4-H

The V850E/IG4-H incorporates a total of 63 input/output ports (including 12 input-only ports) labeled ports 0 to 5, 7, and DL. The port configuration is shown in Figure 4-1.

There are three power supply systems for the I/O buffer of a pin: AV_{DD2}, EV_{DD0}, EV_{DD1}, EV_{DD2}, and UV_{DD}. The relationship between each of these power supplies and the pin is shown in Table 4-1.

Figure 4-1. Port Configuration

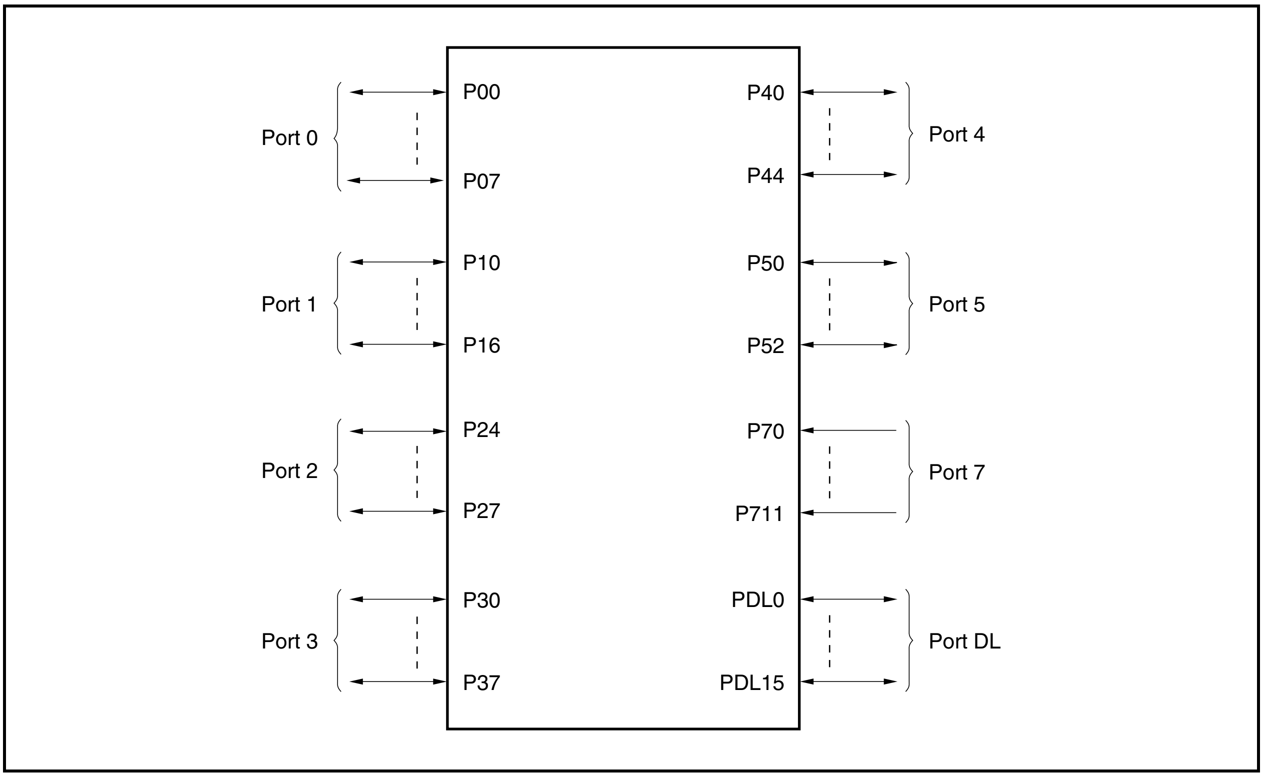


Table 4-1. Power Supplies for I/O Buffer of Each Pin

Power Supply	Corresponding Pins
AV_{DD2}	P70 to P711
EV_{DD0}, EV_{DD1}, EV_{DD2}	P00 to P07, P10 to P16, P24 to P27, P30 to P37, P40 to P44, P50 to P52, PDL0 to PDL15, $\overline{\text{RESET}}$, DCK, DDI, DDO, DMS, $\overline{\text{DRST}}$
UV_{DD}	UDMF, UDPF

4.2.2 V850E/IH4-H

The V850E/IH4-H incorporates a total of 80 input/output ports (including 12 input-only ports) labeled ports 0 to 5, 7, 9, and DL. The port configuration is shown in Figure 4-2.

There are three power supply systems for the I/O buffer of a pin: AV_{DD2}, EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}, and UV_{DD}. The relationship between each of these power supplies and the pin is shown in Table 4-2.

Figure 4-2. Port Configuration

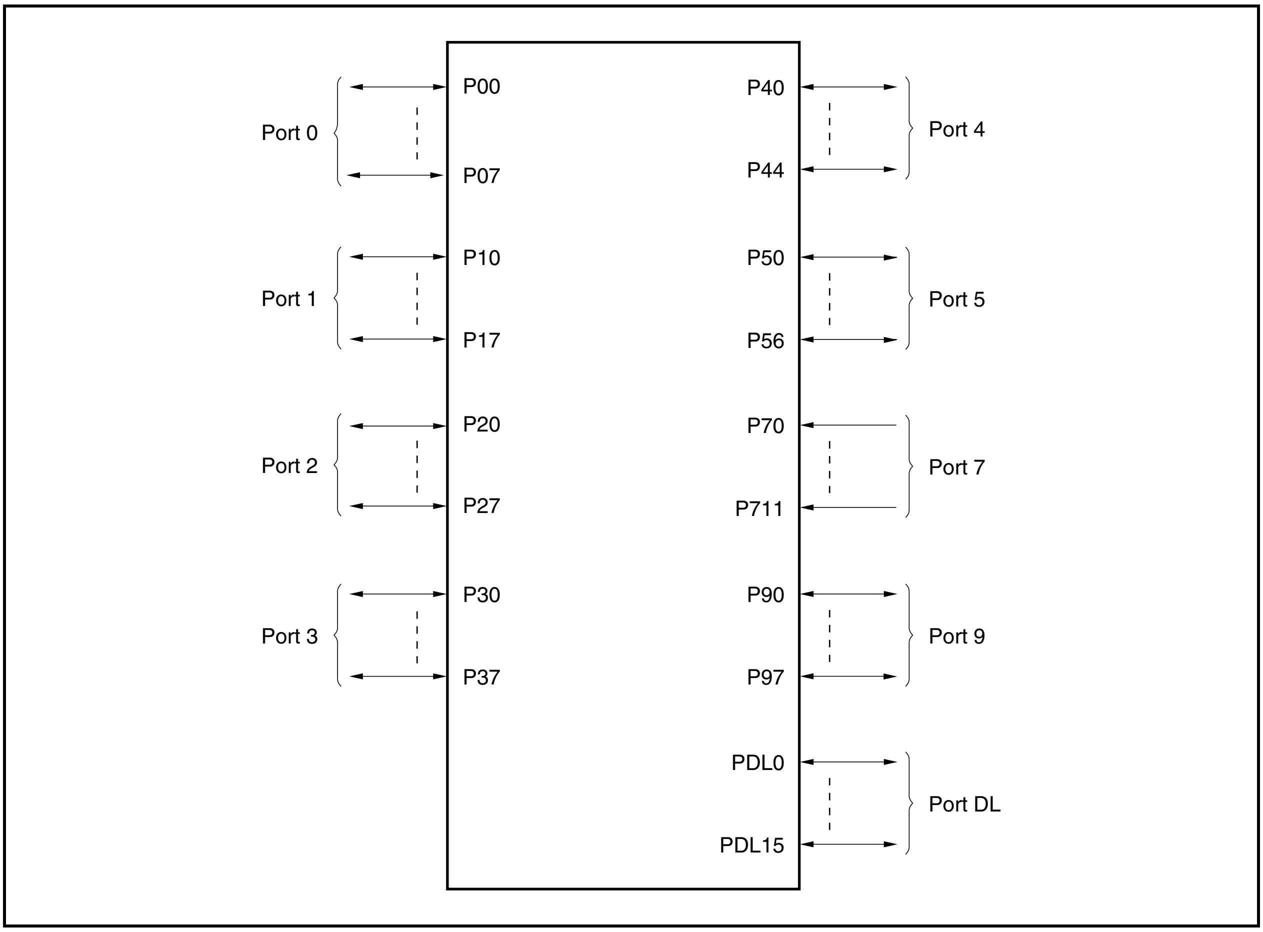


Table 4-2. Power Supplies for I/O Buffer of Each Pin

Power Supply	Corresponding Pins
AV_{DD2}	P70 to P711
EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P44, P50 to P56, P90 to P97, PDL0 to PDL15, $\overline{\text{RESET}}$, DCK, DDI, DDO, DMS, $\overline{\text{DRST}}$, TRCCLK, TRCDATA0 to TRCDATA3, TRCEND
UV_{DD}	UDMF, UDPF

4.3 Port Configuration

Table 4-3. Port Configuration (V850E/IG4-H)

Item	Configuration
Control registers	Port n register (Pn: n = 0 to 5, 7, DL) Port n mode register (PMn: n = 0 to 5, DL) Port n mode control register (PMCn: n = 0 to 5, 7, DL) Port n function control register (PFCn: n = 0 to 5, DL) Port n function control expansion register (PFCEn: n = 0 to 5, DL) Pull-up resistor option register (PUn: n = 0 to 5, DL) Port 3 function register (PF3)
Ports	Input-only: 12, I/O: 51
Pull-up resistor	Software control: 51

Table 4-4. Port Configuration (V850E/IH4-H)

Item	Configuration
Control registers	Port n register (Pn: n = 0 to 5, 7, 9, DL) Port n mode register (PMn: n = 0 to 5, 9, DL) Port n mode control register (PMCn: n = 0 to 5, 7, 9, DL) Port n function control register (PFCn: n = 0 to 5, DL) Port n function control expansion register (PFCEn: n = 0 to 5, DL) Pull-up resistor option register (PUn: n = 0 to 5, 9, DL) Port 3 function register (PF3)
Ports	Input-only: 12, I/O: 68
Pull-up resistor	Software control: 68

(1) Port n register (Pn)

Data is input from or output to an external device by writing or reading the Pn register.

The Pn register consists of a port latch that holds output data, and a circuit that reads the status of pins.

Each bit of the Pn register corresponds to one pin of port n, and can be read or written in 1-bit units.

After reset: Undefined R/W

	7	6	5	7	3	2	1	0
Pn	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pnm	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Data is written to or read from the Pn register as follows, regardless of the setting of the PMCn register.

Table 4-5. Writing/Reading Pn Register

Setting of PMn Register	Writing to Pn Register	Reading from Pn Register
Output mode (PMnm = 0)	Data is written to the output latch[Note 1]. In the port mode (PMCn = 0), the contents of the output latch are output from the pins.	The value of the output latch is read[Note 2].
Input mode (PMnm = 1)	Data is written to the output latch. The pin status is not affected[Note 1].	The pin status is read[Note 3].

Notes 1. The value written to the output latch is retained until a new value is written to the output latch.

2. Also, the value of the Pn register is read when the PMn register is in the output mode while the alternate function is set.

3. If the PMn register is in the input mode while the alternate function is set, the statuses of the pins at that time are read regardless of whether the alternate function is an input or output function.

(2) Port n mode register (PMn)

The PMn register specifies the input or output mode of the corresponding port pin.

Each bit of this register corresponds to one pin of port n, and the input or output mode can be specified in 1-bit units.

After reset: FFH R/W

	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0

PMnm	Control of I/O mode
0	Output mode
1	Input mode

(3) Port n mode control register (PMCn)

The PMCn register specifies the port mode or alternate function.

Each bit of this register corresponds to one pin of port n, and the mode of the port can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PMCn	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0

PMCnm	Specification of operating mode
0	Port mode
1	Alternate function

(4) Port n function control register (PFCn)

The PFCn register specifies the alternate function of a port pin to be used if the pin has two alternate functions.

Each bit of this register corresponds to one pin of port n, and the alternate function of a port pin can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

PFCnm	Specification of alternate function
0	Alternate function 1
1	Alternate function 2

(5) Port n function control expansion register (PFCEn)

The PFCEn register specifies the alternate function of a port pin to be used if the pin has three or more alternate functions.

Each bit of this register corresponds to one pin of port n, and the alternate function of a port pin can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFCEn	PFCEn7	PFCEn6	PFCEn5	PFCEn4	PFCEn3	PFCEn2	PFCEn1	PFCEn0

	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

PFCEnm	PFCnm	Specification of alternate function
0	0	Alternate function 1
0	1	Alternate function 2
1	0	Alternate function 3
1	1	Alternate function 4

(6) Pull-up resistor option register (PUn)

PUn is a register that specifies the connection of an on-chip pull-up resistor.

Each bit of the pull-up resistor option register corresponds to one pin of port n and can be specified in 1-bit units.

After reset: 00H R/W

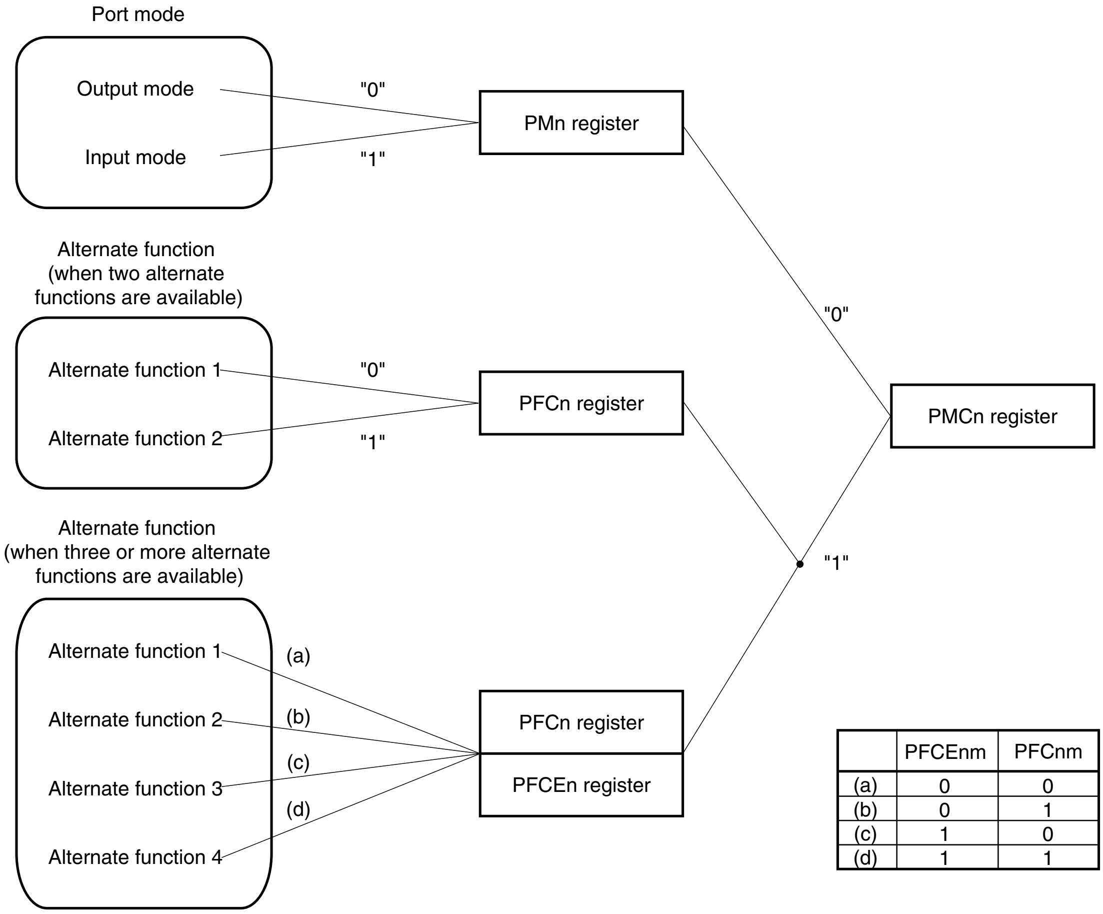
	7	6	5	4	3	2	1	0
PUn	PUn7	PUn6	PUn5	PUn4	PUn3	PUn2	PUn1	PUn0

PUnm	Control of on-chip pull-up resistor connection
0	Not connected
1	Connected

(7) Port settings

Set the ports as follows.

Figure 4-3. Register Settings and Pin Functions



	PFCEnm	PFCnm
(a)	0	0
(b)	0	1
(c)	1	0
(d)	1	1

Caution To switch to external interrupt input (INTPn) from the port mode (by changing the PMCa.PMCam bit from 0 to 1), an external interrupt may be input if a wrong valid edge is detected. Therefore, be sure to set “no edge detection” by INTRk, INTFk, ADTR, or ADTF register, select external interrupt input (INTPn), and then specify the valid edge (n = 00 to 19, ADT0, ADT1, a = 0 to 5, DL, m = 0 to 7, k = 0 to 3).

When switching to the port mode from external interrupt input (INTPn) (by changing the PMCam bit = from 1 to 0), an edge may be detected. Therefore, be sure to set “no edge detection” by INTRk, INTFk, ADTR, or ADTF register, and then select the port mode.

Remark Switch to the alternate function using the following procedure (for n, see **Tables 4-3** and **4-4**).

<1> Set the PFCn and PFCEn registers.

<2> Set the PMCn register.

<3> Set the INTRk, INTFk, ADTR, and ADTF registers (when external interrupt pin is set).

If the PMCn register is set before setting the PFCn and PFCEn registers, an unexpected peripheral function may be selected while the PFCn and PFCEn registers are being set.

4.3.1 Port 0

Port 0 can be set to the input or output mode in 1-bit units.

The pins of port 0 have the following alternate functions.

Table 4-6. Alternate Functions of Port 0

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note]
	IG4-H	IH4-H			
	GC	GF			
P00	89	26	TECR0/TIT00/TOT00/INTP00	I/O	Provided
P01	88	25	TENC00/EVTT0/INTP01	Input	
P02	87	24	TENC01/TIT01/TOT01/INTP02	I/O	
P03	86	23	TOT20/TIT20/TOT2OFF/INTP03	I/O	
P04	85	22	TOT21/TIT21/INTP04	I/O	
P05	84	21	TOT30/TIT30/TOT3OFF/INTP05	I/O	
P06	83	20	TOT31/TIT31/INTP06	I/O	
P07	82	19	TOB01OFF/INTP07/CLKOUT	I/O	

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

Note Software pull-up function

Cautions 1. To control the high-impedance output of a timer for motor control, be sure to set the PMC0.PMC0n bit to 1 and then specify the edge to be detected and enable the operation of the high-impedance output controller, because the output of the motor control timer may go into a high-impedance state if a wrong valid edge is detected (n = 3, 5).

2. When P04 and P06 are used as TOT21 and TOT31, they go into a high-impedance state by inputting the following active signal.

- **Output of high impedance setting signal from high impedance output controller**
- **Output of clock stop detection signal from clock monitor**

3. To switch to external interrupt input (INTP0n) from the port mode (by changing the PMC0.PMC0n bit from 0 to 1), an external interrupt may be input if a wrong valid edge is detected. Therefore, be sure to disable edge detection (INTF0.INTF0n bit = 0 and INTR0.INTR0n bit = 0), select external interrupt input (INTP0n), and then specify the valid edge (n = 0 to 7).

When switching to the port mode from external interrupt input (INTP0n) (by changing the PMC0n bit from 1 to 0), an edge may be detected. Therefore, be sure to disable edge detection (INTF0n bit = 0, INTR0n bit = 0), and then select the port mode.

4. To control high-impedance output of the external interrupt function and motor output control function, set the PMC0n bit to 1 (n = 0 to 7).

(1) Registers

(a) Port 0 register (P0)

After reset: Undefined R/W Address: FFFFF400H

	7	6	5	4	3	2	1	0
P0	P07	P06	P05	P04	P03	P02	P01	P00

P0n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 7

(b) Port 0 mode register (PM0)

After reset: FFH R/W Address: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 7

(c) Port 0 mode control register (PMC0)

After reset: 00H R/W Address: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00

PMC07	Specification of operating mode of P07 pin
0	I/O port
1	TOB01OFF input/INTP07 input/CLKOUT output

PMC06	Specification of operating mode of P06 pin
0	I/O port
1	TOT31 output/TIT31 input/INTP06 input

PMC05	Specification of operating mode of P05 pin
0	I/O port
1	TOT30 output/TIT30 input/TOT3OFF input/INTP05 input

PMC04	Specification of operating mode of P04 pin
0	I/O port
1	TOT21 output/TIT21 input/INTP04 input

PMC03	Specification of operating mode of P03 pin
0	I/O port
1	TOT20 output/TIT20 input/TOT2OFF input/INTP03 input

PMC02	Specification of operating mode of P02 pin
0	I/O port
1	TENC01 input/TIT01 input/TOT01 output/INTP02 input

PMC01	Specification of operating mode of P01 pin
0	I/O port
1	TENC00 input/EVTT0 input/INTP01 input

PMC00	Specification of operating mode of P00 pin
0	I/O port
1	TECR0 input/TIT00 input/TOT00 output/INTP00 input

(d) Port 0 function control register (PFC0)

After reset: 00H R/W Address: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	PFC07	PFC06	PFC05	PFC04	PFC03	PFC02	PFC01	PFC00

Remark For the specifications of alternate functions, see **4.3.1 (1) (f) Settings of alternate functions of port 0**.

(e) Port 0 function control expansion register (PFCE0)

After reset: 00H R/W Address: FFFFF700H

	7	6	5	4	3	2	1	0
PFCE0	0	PFCE06	PFCE05	PFCE04	PFCE03	PFCE02	PFCE01	PFCE00

Remark For the specifications of alternate functions, see **4.3.1 (1) (f) Settings of alternate functions of port 0**.

(f) Setting of alternate function of port 0

PFC07	Specification of Alternate Function of P07 Pin
0	TOB01OFF input/INTP07 input (two functions are alternately used)
1	CLKOUT output

PFCE06	PFC06	Specification of Alternate Function of P06 Pin
0	0	TOT31 output
0	1	TIT31 input
1	0	INTP06 input
1	1	Setting prohibited

PFCE05	PFC05	Specification of Alternate Function of P05 Pin
0	0	TOT30 output
0	1	TIT30 input
1	0	TOT3OFF input/INTP05 input (two functions are alternately used)
1	1	Setting prohibited

PFCE04	PFC04	Specification of Alternate Function of P04 Pin
0	0	TOT21 output
0	1	TIT21 input
1	0	INTP04 input
1	1	Setting prohibited

PFCE03	PFC03	Specification of Alternate Function of P03 Pin
0	0	TOT20 output
0	1	TIT20 input
1	0	TOT2OFF input/INTP03 input (two functions are alternately used)
1	1	Setting prohibited

PFCE02	PFC02	Specification of Alternate Function of P02 Pin
0	0	TENC01 input/TIT01 input (two functions are alternately used)
0	1	TOT01 output
1	0	INTP02 input
1	1	Setting prohibited

PFCE01	PFC01	Specification of Alternate Function of P01 Pin
0	0	TENC00 input
0	1	EVTT0 input
1	0	INTP01 input
1	1	Setting prohibited

PFCE00	PFC00	Specification of Alternate Function of P00 Pin
0	0	TECR0 input/TIT00 input (two functions are alternately used)
0	1	TOT00 output
1	0	INTP00 input
1	1	Setting prohibited

(g) Pull-up resistor option register 0 (PU0)

After reset: 00H R/W Address: FFFFFC40H

	7	6	5	4	3	2	1	0
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00

PU0n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOT21 and TOT31 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOT2OFF or TOA3OFF pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 7

4.3.2 Port 1

Port 1 can be set to the input or output mode in 1-bit units.

The number of I/O pins for port 1 differs depending on the product.

Generic Name	Number of I/O Ports
V850E/IG4-H	7-bit I/O port
V850E/IH4-H	8-bit I/O port

The pins of port 1 have the following alternate functions.

Table 4-7. Alternate Functions of Port 1

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IG4-H	IH4-H			
	GC	GF			
P10	98	36	TOB0T1/TIB01/TOB01	I/O	Provided
P11	97	35	TOB0B1/TIB02/TOB02	I/O	
P12	96	34	TOB0T2/TIB03/TOB03	I/O	
P13	95	33	TOB0B2/TIB00	I/O	
P14	94	32	TOB0T3/EVTB0	I/O	
P15	93	31	TOB0B3/TRGB0	I/O	
P16	92	30	TOB00/TOB0OFF/INTP08/ADTRG0/INTADT0	I/O	
P17[Note 2]	–	29	–	–	

Notes 1. Software pull-up function

2. V850E/IH4-H only

Caution **When P10 to P15 are used as TOB0T1 to TOB0T3 and TOB0B1 to TOB0B3, they go into a high-impedance state by inputting the following active signal.**

- **Output of high impedance setting signal from high impedance output controller**
- **Output of clock stop detection signal from clock monitor**

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(1) Registers

(a) Port 1 register (P1)

After reset: Undefined R/W Address: FFFFF402H

	7	6	5	4	3	2	1	0
P1	P17[Note]	P16	P15	P14	P13	P12	P11	P10

P1n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Note Valid only in the V850E/IH4-H.
For the V850E/IG4-H, the read value of this bit is undefined.

Remark V850E/IG4-H: n = 0 to 6
V850E/IH4-H: n = 0 to 7

(b) Port 1 mode register (PM1)

After reset: FFH R/W Address: FFFFF422H

	7	6	5	4	3	2	1	0
PM1	PM17[Note]	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Note Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 1.

Remark V850E/IG4-H: n = 0 to 6
V850E/IH4-H: n = 0 to 7

(c) Port 1 mode control register (PMC1)

After reset: 00H R/W Address: FFFFF442H

	7	6	5	4	3	2	1	0
PMC1	0	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10

PMC16	Specification of operating mode of P16 pin
0	I/O port
1	TOB00 output/TOB0OFF input/INTP08 input/ADTRG0 input/INTADT0 input

PMC15	Specification of operating mode of P15 pin
0	I/O port
1	TOB0B3 output/TRGB0 input

PMC14	Specification of operating mode of P14 pin
0	I/O port
1	TOB0T3 output/EVTB0 input

PMC13	Specification of operating mode of P13 pin
0	I/O port
1	TOB0B2 output/TIB00 input

PMC12	Specification of operating mode of P12 pin
0	I/O port
1	TOB0T2 output/TIB03 input/TOB03 output

PMC11	Specification of operating mode of P11 pin
0	I/O port
1	TOB0B1 output/TIB02 input/TOB02 output

PMC10	Specification of operating mode of P10 pin
0	I/O port
1	TOB0T1 output/TIB01 input/TOB01output

(d) Port 1 function control register (PFC1)

After reset: 00H R/W Address: FFFFF462H

	7	6	5	4	3	2	1	0
PFC1	0	PFC16	PFC15	PFC14	PFC13	PFC12	PFC11	PFC10

Remark For the specifications of alternate functions, see **4.3.2 (1) (f) Settings of alternate functions of port 1**.

(e) Port 1 function control expansion register (PFCE1)

After reset: 00H R/W Address: FFFFF702H

	7	6	5	4	3	2	1	0
PFCE1	0	PFCE16	0	0	0	PFCE12	PFCE11	PFCE10

Remark For the specifications of alternate functions, see **4.3.2 (1) (f) Settings of alternate functions of port 1**.

(f) Settings of alternate functions of port 1

PFCE16	PFC16	Specification of Alternate Function of P16 Pin
0	0	TOB00 output
0	1	TOB0OFF input/INTP08 input (two functions are alternately used)
1	0	ADTRG0 input/INTADT0 input (two functions are alternately used)
1	1	Setting prohibited

PFC15	Specification of Alternate Function of P15 Pin
0	TOB0B3 output
1	TRGB0 input

PFC14	Specification of Alternate Function of P14 Pin
0	TOB0T3 output
1	EVTB0 input

PFC13	Specification of Alternate Function of P13 Pin
0	TOB0B2 output
1	TIB00 input

PFCE12	PFC12	Specification of Alternate Function of P12 Pin
0	0	TOB0T2 output
0	1	TIB03 input
1	0	TOB03 output
1	1	Setting prohibited

PFCE11	PFC11	Specification of Alternate Function of P11 Pin
0	0	TOB0B1 output
0	1	TIB02 input
1	0	TOB02 output
1	1	Setting prohibited

PFCE10	PFC10	Specification of Alternate Function of P10 Pin
0	0	TOB0T1 output
0	1	TIB01 input
1	0	TOB01 output
1	1	Setting prohibited

(g) Pull-up resistor option register 1 (PU1)

After reset: 00H R/W Address: FFFFFC42H

	7	6	5	4	3	2	1	0
PU1	PU17[Note 1]	PU16	PU15	PU14	PU13	PU12	PU11	PU10

PU1n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note 2]

Notes 1. Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 0.

2. An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3 and TOB0B1 to TOB0B3 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOB01OFF pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark V850E/IG4-H: n = 0 to 6
V850E/IH4-H: n = 0 to 7

4.3.3 Port 2

Port 2 can be set to the input or output mode in 1-bit units.

The number of I/O pins for port 2 differs depending on the product.

Generic Name	Number of I/O Ports
V850E/IG4-H	4-bit I/O port
V850E/IH4-H	8-bit I/O port

The pins of port 2 have the following alternate functions.

Table 4-8. Alternate Functions of Port 2

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IG4-H	IH4-H			
	GC	GF			
P20[Note 2]	–	67	TOB1T1[Note 2]/TIB11[Note 2]/TOB11[Note 2]	I/O	Provided
P21[Note 2]	–	68	TOB1B1[Note 2]/TIB12[Note 2]/TOB12[Note 2]	I/O	
P22[Note 2]	–	69	TOB1T2[Note 2]/TIB13[Note 2]/TOB13[Note 2]	I/O	
P23[Note 2]	–	70	TOB1B2[Note 2]/TIB10	I/O	
P24	28	71	TOB1T3/EVTB1	I/O	
P25	29	72	TOB1B3/TRGB1	I/O	
P26	30	73	TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1	I/O	
P27	43	87	INTP09/$\overline{\text{WR0}}$/TOA01	I/O	

Notes **1.** Software pull-up function

2. V850E/IH4-H only

Caution When P20 to P25 are used as TOB1T1 (V850E/IH4-H only), TOB1T2 (V850E/IH4-H only), TOB1T3, TOB1B1 (V850E/IH4-H only), TOB1B2 (V850E/IH4-H only), and TOB1B3, they go into a high-impedance state by inputting the following active signal.

- **Output of high impedance setting signal from high impedance output controller**
- **Output of clock stop detection signal from clock monitor**

Remark IG4-H: V850E/IG4-H

IH4-H: V850E/IH4-H

GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(1) Registers

(a) Port 2 register (P2)

After reset: Undefined R/W Address: FFFFF404H

	7	6	5	4	3	2	1	0
P2	P27	P26	P25	P24	P23[Note]	P22[Note]	P21[Note]	P20[Note]

P2n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Note Valid only in the V850E/IH4-H.
For the V850E/IG4-H, the read value of this bit is undefined.

Remark V850E/IG4-H: n = 4 to 7
V850E/IH4-H: n = 0 to 7

(b) Port 2 mode register (PM2)

After reset: FFH R/W Address: FFFFF424H

	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23[Note]	PM22[Note]	PM21[Note]	PM20[Note]

PM2n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Note Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 1.

Remark V850E/IG4-H: n = 4 to 7
V850E/IH4-H: n = 0 to 7

(c) Port 2 mode control register (PMC2)

After reset: 00H R/W Address: FFFFF444H

	7	6	5	4	3	2	1	0
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23[Note 1]	PMC22[Note 1]	PMC21[Note 1]	PMC20[Note 1]

PMC27	Specification of operating mode of P27 pin
0	I/O port
1	INTP09 input/$\overline{\text{WR0}}$ output/TOA01 output

PMC26	Specification of operating mode of P26 pin
0	I/O port
1	TOB10 output/TOB1OFF input/INTP10 input/ADTRG1 input/INTADT1 input

PMC25	Specification of operating mode of P25 pin
0	I/O port
1	TOB1B3 output/TRGB1 input

PMC24	Specification of operating mode of P24 pin
0	I/O port
1	TOB1T3 output/EVTB1 input

PMC23[Note 1]	Specification of operating mode of P23 pin
0	I/O port
1	TOB1B2 output[Note 2]/TIB10 input[Note 2]

PMC22[Note 1]	Specification of operating mode of P22 pin
0	I/O port
1	TOB1T2 output[Note 2]/TIB13 input[Note 2]/TOB13 output[Note 2]

PMC21[Note 1]	Specification of operating mode of P21 pin
0	I/O port
1	TOB1B1 output[Note 2]/TIB12 input[Note 2]/TOB12 output[Note 2]

PMC20[Note 1]	Specification of operating mode of P20 pin
0	I/O port
1	TOB1T1 output[Note 2]/TIB11 input[Note 2]/TOB11 output[Note 2]

Notes 1. Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 0.

2. V850E/IH4-H only

(d) Port 2 function control register (PFC2)

After reset: 00H R/W Address: FFFFF464H

	7	6	5	4	3	2	1	0
PFC2	PFC27	PFC26	PFC25	PFC24	PFC23[Note]	PFC22[Note]	PFC21[Note]	PFC20[Note]

Note Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 0.

Remark For the specifications of alternate functions, see **4.3.3 (1) (f) Settings of alternate functions of port 2**.

(e) Port 2 function control expansion register (PFCE2)

After reset: 00H R/W Address: FFFFF704H

	7	6	5	4	3	2	1	0
PFCE2	PFCE27	PFCE26	0	0	0	PFCE22[Note]	PFCE21[Note]	PFCE20[Note]

Note Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 0.

Remark For the specifications of alternate functions, see **4.3.3 (1) (f) Settings of alternate functions of port 2**.

(f) Settings of alternate functions of port 2

PFCE27	PFC27	Specification of Alternate Function of P27 Pin
0	0	INTP09 input
0	1	$\overline{\text{WR0}}$ output
1	0	TOA01 output
1	1	Setting prohibited

PFCE26	PFC26	Specification of Alternate Function of P26 Pin
0	0	TOB10 output
0	1	TOB1OFF input/INTP10 input (two functions are alternately used)
1	0	ADTRG1 input/INTADT1 input (two functions are alternately used)
1	1	Setting prohibited

PFC25	Specification of Alternate Function of P25 Pin
0	TOB1B3 output
1	TRGB1 input

PFC24	Specification of Alternate Function of P24 Pin
0	TOB1T3 output
1	EVTB1 input

PFC23[Note 1]	Specification of Alternate Function of P23 Pin
0	TOB1B2 output[Note 2]
1	TIB10 input[Note 2]

PFCE22[Note 1]	PFC22[Note 1]	Specification of Alternate Function of P22 Pin
0	0	TOB1T2 output[Note 2]
0	1	TIB13 input[Note 2]
1	0	TOB13 output[Note 2]
1	1	Setting prohibited

PFCE21[Note 1]	PFC21[Note 1]	Specification of Alternate Function of P21 Pin
0	0	TOB1B1 output[Note 2]
0	1	TIB12 input[Note 2]
1	0	TOB12 output[Note 2]
1	1	Setting prohibited

PFCE20[Note 1]	PFC20[Note 1]	Specification of Alternate Function of P20 Pin
0	0	TOB1T1 output[Note 2]
0	1	TIB11 input[Note 2]
1	0	TOB11 output[Note 2]
1	1	Setting prohibited

Notes 1. Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 0.

2. Valid only in the V850E/IH4-H.
For the V850E/IG4-H, setting prohibited.

(g) Pull-up resistor option register 2 (PU2)

After reset: 00H R/W Address: FFFFFC44H

	7	6	5	4	3	2	1	0
PU2	PU27	PU26	PU25	PU24	PU23[Note 1]	PU22[Note 1]	PU21[Note 1]	PU20[Note 1]

PU2n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note 2]

Notes 1. Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 0.

2. An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB1T1 (V850E/IH4-H only), TOB1T2 (V850E/IH4-H only), TOB1T3, TOB1B1 (V850E/IH4-H only), TOB1B2 (V850E/IH4-H only), and TOB1B3 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB1OFF or TOB01OFF pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark V850E/IG4-H: n = 4 to 7
V850E/IH4-H: n = 0 to 7

4.3.4 Port 3

Port 3 can be set to the input or output mode in 1-bit units.

The pins of port 3 have the following alternate functions.

Table 4-9. Alternate Functions of Port 3

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note]
	IG4-H	IH4-H			
	GC	GF			
P30	54	106	RXDA1/SCL/$\overline{WR1}$	I/O	Provided
P31	55	107	TXDA1/SDA/$\overline{WAIT}$	I/O	
P32	56	108	SIF1/RXDA2/$\overline{CS1}$	I/O	
P33	57	109	SOF1/TXDA2	Output	
P34	58	110	$\overline{SCKF1}$/INTP11/$\overline{CS0}$	I/O	
P35	59	111	SIF2/RXDB	Input	
P36	60	112	SOF2/TXDB	Output	
P37	61	113	$\overline{SCKF2}$/INTP12/ASTB	I/O	

Note Software pull-up function

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(1) Registers

(a) Port 3 register (P3)

After reset: Undefined R/W Address: FFFFF406H

	7	6	5	4	3	2	1	0
P3	P37	P36	P35	P34	P33	P32	P31	P30

P3n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 7

(b) Port 3 mode register (PM3)

After reset: FFH R/W Address: FFFFF426H

	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 7

(c) Port 3 mode control register (PMC3)

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC37	Specification of operating mode of P37 pin
0	I/O port
1	$\overline{\text{SCKF2}}$ I/O/INTP12 input/ASTB output

PMC36	Specification of operating mode of P36 pin
0	I/O port
1	SOF2 output/TXDB output

PMC35	Specification of operating mode of P35 pin
0	I/O port
1	SIF2 input/RXDB input

PMC34	Specification of operating mode of P34 pin
0	I/O port
1	$\overline{\text{SCKF1}}$ I/O/INTP11 input/$\overline{\text{CS0}}$ output

PMC33	Specification of operating mode of P33 pin
0	I/O port
1	SOF1 output/TXDA2 output

PMC32	Specification of operating mode of P32 pin
0	I/O port
1	SIF1 input/RXDA2 input/$\overline{\text{CS1}}$ output

PMC31	Specification of operating mode of P31 pin
0	I/O port
1	TXDA1 output/SDA I/O/$\overline{\text{WAIT}}$ input

PMC30	Specification of operating mode of P30 pin
0	I/O port
1	RXDA1 input/SCL I/O/$\overline{\text{WR1}}$ output

(d) Port 3 function control register (PFC3)

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

Remark For the specifications of alternate functions, see **4.3.4 (1) (f) Settings of alternate functions of port 3**.

(e) Port 3 function control expansion register (PFCE3)

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	0	0	PFCE34	0	PFCE32	PFCE31	PFCE30

Remark For the specifications of alternate functions, see **4.3.4 (1) (f) Settings of alternate functions of port 3**.

(f) Settings of alternate functions of port 3

PFCE37	PFC37	Specification of Alternate Function of P37 Pin
0	0	$\overline{\text{SCKF2}}$ input/output
0	1	INTP12 input
1	0	ASTB output
1	1	Setting prohibited

PFC36	Specification of Alternate Function of P36 Pin
0	SOF2 output
1	TXDB output

PFC35	Specification of Alternate Function of P35 Pin
0	SIF2 input
1	RXDB input

PFCE34	PFC34	Specification of Alternate Function of P34 Pin
0	0	$\overline{\text{SCKF1}}$ input/output
0	1	INTP11 input
1	0	$\overline{\text{CS0}}$ output
1	1	Setting prohibited

PFC33	Specification of Alternate Function of P33 Pin
0	SOF1 output
1	TXDA2 output

PFCE32	PFC32	Specification of Alternate Function of P32 Pin
0	0	SIF1 input
0	1	RXDA2 input
1	0	$\overline{\text{CS1}}$ output
1	1	Setting prohibited

PFCE31	PFC31	Specification of Alternate Function of P31 Pin
0	0	TXDA1 output
0	1	SDA input/output
1	0	$\overline{\text{WAIT}}$ input
1	1	Setting prohibited

PFCE30	PFC30	Specification of Alternate Function of P30 Pin
0	0	RXDA1 input
0	1	SCL input/output
1	0	$\overline{\text{WR1}}$ output
1	1	Setting prohibited

(g) Pull-up resistor option register 3 (PU3)

After reset: 00H R/W Address: FFFFFC46H

	7	6	5	4	3	2	1	0
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode (including the $\overline{\text{SCKF1}}$ and $\overline{\text{SCKF2}}$ pins in the slave mode). An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 7

(h) Port 3 function register (PF3)

After reset: 00H R/W Address: FFFFFC66H

	7	6	5	4	3	2	1	0
PF3	0	0	0	0	0	0	PF31	PF30

PF3n	Control of normal output/N-ch open-drain output
0	Normal output (CMOS output)
1	N-ch open-drain output[Note]

Note When using I^{2}C, set as N-ch open-drain output.

Remark n = 0, 1

4.3.5 Port 4

Port 4 can be set to the input or output mode in 1-bit units.

The pins of port 4 have the following alternate functions.

Table 4-10. Alternate Functions of Port 4

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IG4-H	IH4-H			
	GC	GF			
P40	46	96	SIF0/RXDA0/DDI[Note 2]/TOA00	I/O	Provided
P41	47	97	SOF0/TXDA0	Output	
P42	48	98	$\overline{\text{SCKF0}}$/DCK[Note 2]/TOA10	I/O	
P43	49	99	INTP13/DMS[Note 2]/TOA11	I/O	
P44	50	100	INTP14/$\overline{\text{RD}}$	I/O	

Notes 1. Software pull-up function

2. The P40, P42, and P43 pins are also used for on-chip debugging. Switching between the on-chip debug function and port function (including the alternate function) can be done by using the $\overline{\text{DRST}}$ pin level. The following shows the setting method.

Port 4 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P40/SIF0/RXDA0/TOA00	DDI
P42/$\overline{\text{SCKF0}}$/TOA10	DCK
P43/INTP13/TOA11	DMS

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(1) Registers

(a) Port 4 register (P4)

After reset: Undefined R/W Address: FFFFF408H

	7	6	5	4	3	2	1	0
P4	0	0	0	P44	P43	P42	P41	P40

P4n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 4

(b) Port 4 mode register (PM4)

After reset: FFH R/W Address: FFFFF428H

	7	6	5	4	3	2	1	0
PM4	0	0	0	PM44	PM43	PM42	PM41	PM40

PM4n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 4

(c) Port 4 mode control register (PMC4)

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	0	0	PMC44	PMC43	PMC42	PMC41	PMC40

PMC44	Specification of operating mode of P44 pin
0	I/O port
1	INTP14 input/$\overline{\text{RD}}$ output

PMC43	Specification of operating mode of P43 pin
0	I/O port
1	INTP13 input/TOA11 output

PMC42	Specification of operating mode of P42 pin
0	I/O port
1	$\overline{\text{SCKF0}}$ I/O/TOA10 output

PMC41	Specification of operating mode of P41 pin
0	I/O port
1	SOF0 output/TXDA0 output

PMC40	Specification of operating mode of P40 pin
0	I/O port
1	SIF0 input/RXDA0 input/TOA00 output

(d) Port 4 function control register (PFC4)

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	PFC44	PFC43	0	PFC41	PFC40

Remark For the specifications of alternate functions, see **4.3.5 (1) (f) Settings of alternate functions of port 4**.

(e) Port 4 function control expansion register (PFCE4)

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	0	0	0	0	0	PFCE42	0	PFCE40

Remark For the specifications of alternate functions, see **4.3.5 (1) (f) Settings of alternate functions of port 4**.

(f) Settings of alternate functions of port 4

PFC44	Specification of Alternate Function of P44 Pin
0	INTP14 input
1	$\overline{\text{RD}}$ output

PFC43	Specification of Alternate Function of P43 Pin
0	INTP13 input
1	TOA11 output

PFCE42	Specification of Alternate Function of P42 Pin
0	$\overline{\text{SCKF0}}$ input/output
1	TOA10 output

PFC41	Specification of Alternate Function of P41 Pin
0	SOF0 output
1	TXDA0 output

PFCE40	PFC40	Specification of Alternate Function of P40 Pin
0	0	SIF0 input
0	1	RXDA0 input
1	0	Setting prohibited
1	1	TOA00 output

(g) Pull-up resistor option register 4 (PU4)

After reset: 00H R/W Address: FFFFFC48H

	7	6	5	4	3	2	1	0
PU4	0	0	0	PU44	PU43	PU42	PU41	PU40

PU4n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode (including the $\overline{\text{SCKF0}}$ pin in the slave mode). An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 4

4.3.6 Port 5

Port 5 can be set to the input or output mode in 1-bit units.

The number of I/O pins for port 5 differs depending on the product.

Generic Name	Number of I/O Ports
V850E/IG4-H	3-bit I/O port
V850E/IH4-H	7-bit I/O port

The pins of port 5 have the following alternate functions.

Table 4-11. Alternate Functions of Port 5

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IG4-H	IH4-H			
	GC	GF			
P50	51	103	TECR1/TIT10/TOT10/INTP17	I/O	Provided
P51	52	104	TENC10/EVTT1/INTP18/UCLK[Note 2]	Input	
P52	53	105	TENC11/TIT11/TOT11/INTP19	I/O	
P53[Note 3]	–	101	UCLK[Note 3]	Input	
P54[Note 3]	–	102	–	–	
P55[Note 3]	–	10	–	–	
P56[Note 3]	–	9	–	–	

Notes 1. Software pull-up function

2. V850E/IG4-H only

3. V850E/IH4-H only

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(1) Registers

(a) Port 5 register (P5)

After reset: Undefined R/W Address: FFFFF40AH

	7	6	5	4	3	2	1	0
P5	0	P56[Note]	P55[Note]	P54[Note]	P53[Note]	P52	P51	P50

P5n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Note Valid only in the V850E/IH4-H.
For the V850E/IG4-H, the read value of this register is undefined.

Remark V850E/IG4-H: n = 0 to 2
V850E/IH4-H: n = 0 to 6

(b) Port 5 mode register (PM5)

After reset: FFH R/W Address: FFFFF42AH

	7	6	5	4	3	2	1	0
PM5	0	PM56[Note]	PM55[Note]	PM54[Note]	PM53[Note]	PM52	PM51	PM50

PM5n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Note Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set these bits to 1.

Remark V850E/IG4-H: n = 0 to 2
V850E/IH4-H: n = 0 to 6

(c) Port 5 mode control register (PMC5)

After reset: 00H R/W Address: FFFFF44AH

	7	6	5	4	3	2	1	0
PMC5	0	0	0	0	PMC53[Note 1]	PMC52	PMC51	PMC50

PMC53[Note 1]	Specification of operating mode of P53 pin
0	I/O port
1	UCLK input[Note 2]

PMC52	Specification of operating mode of P52 pin
0	I/O port
1	TENC11 input/TIT11 input/TOT11 output/INTP19 input

PMC51	Specification of operating mode of P51 pin
0	I/O port
1	TENC10 input/EVTT1 input/INTP18 input/UCLK input[Note 3]

PMC50	Specification of operating mode of P50 pin
0	I/O port
1	TECR1 input/TIT10 output/TOT10 output/INTP17 input

Notes 1. Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set this bit to 0.

2. Valid only in the V850E/IH4-H.
For the V850E/IG4-H, setting prohibited.

3. V850E/IG4-H only.

(d) Port 5 function control register (PFC5)

After reset: 00H R/W Address: FFFFF46AH

	7	6	5	4	3	2	1	0
PFC5	0	0	0	0	0	PFC52	PFC51	PFC50

Remark For the specifications of alternate functions, see **4.3.6 (1) (f) Settings of alternate functions of port 5**.

(e) Port 5 function control expansion register (PFCE5)

After reset: 00H R/W Address: FFFFF70AH

	7	6	5	4	3	2	1	0
PFCE5	0	0	0	0	0	PFCE52	PFCE51	PFCE50

Remark For the specifications of alternate functions, see **4.3.6 (1) (f) Settings of alternate functions of port 5**.

(f) Settings of alternate functions of port 5

PFCE52	PFC52	Specification of Alternate Function of P52 Pin
0	0	TENC11 input/TIT11 input (two functions are alternately used)
0	1	TOT11 output
1	0	INTP19 input
1	1	Setting prohibited

PFCE51	PFC51	Specification of Alternate Function of P51 Pin
0	0	TENC10 input
0	1	EVTT1 input
1	0	INTP18 input
1	1	UCLK input[Note]

PFCE50	PFC50	Specification of Alternate Function of P50 Pin
0	0	TECR1 input/TIT10 input (two functions are alternately used)
0	1	TOT10 output
1	0	INTP17 input
1	1	Setting prohibited

Note Valid only in the V850E/IG4-H.
For the V850E/IH4-H, setting prohibited.

(g) Pull-up resistor option register 5 (PU5)

After reset: 00H R/W Address: FFFFFC4AH

	7	6	5	4	3	2	1	0
PU5	0	PU56[Note 1]	PU55[Note 1]	PU54[Note 1]	PU53[Note 1]	PU52	PU51	PU50

PU4n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note 2]

Notes 1. Valid only in the V850E/IH4-H.
For the V850E/IG4-H, be sure to set these bits to 0.

2. An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark V850E/IG4-H: n = 0 to 2
V850E/IH4-H: n = 0 to 6

4.3.7 Port 7

Port 7 is an input port with all its pins fixed to the input mode.

The pins of port 7 have the following alternate functions.

Table 4-12. Alternate Functions of Port 7

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note]
	IG4-H	IH4-H			
	GC	GF			
P70	14	53	ANI20	Input	None
P71	15	54	ANI21	Input	
P72	16	55	ANI22	Input	
P73	17	56	ANI23	Input	
P74	18	57	ANI24	Input	
P75	19	58	ANI25	Input	
P76	20	59	ANI26	Input	
P77	21	60	ANI27	Input	
P78	22	61	ANI28	Input	
P79	23	62	ANI29	Input	
P710	24	63	ANI210	Input	
P711	25	64	ANI211	Input	

Note Software pull-up function

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(1) Registers

(a) Port 7 register H, port 7 register L (P7H, P7L)

After reset: Undefined R Address: R7L FFFFFBB0H, R7H FFFFFBB1H

	7	6	5	4	3	2	1	0
P7H	0	0	0	0	P711	P710	P79	P78

	7	6	5	4	3	2	1	0
P7L	P77	P76	P75	P74	P73	P72	P71	P70

P7n	Control of input data
0	Input low level.
1	Input high level.

Caution When using a port input pin and analog input pin (ANI2n) together, be sure to set the bit (PMC7n) of the PMC7 register to be used as the ANI2n pin to 1.

Remark n = 0 to 11

(b) Port 7 mode control register H, port 7 mode control register L (PMC7H, PMC7L)

After reset: 00H R/W Address: PMC7L FFFFFBB8H, PMC7H FFFFFBB9H

	7	6	5	4	3	2	1	0
PMC7H	0	0	0	0	PMC711	PMC710	PMC79	PMC78

	7	6	5	4	3	2	1	0
PMC7L	PMC77	PMC76	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70

PMC7n	Specification of operating mode of P7n pin
0	Input port (reading P7n enabled. Input buffer is on when this bit is read)
1	ANI2n input (reading P7n disabled. Input buffer is off when this bit is read)

Cautions 1. Do not change to the port mode using A/D converter 2 during A/D conversion.

2. The PMC7H and PMC7L registers enable or disable reading of the P7H and P7L registers, respectively. When the PMC7n bit is 1, the input buffer does not turn on even when the P7H and P7L registers are read. In this case, the read value of the P7n bit is fixed to the low level. This is to prevent through-current that may flow when the ANI2n input (intermediate level) is read.

Remark n = 0 to 11

4.3.8 Port 9 (V850E/IH4-H only)

Port 9 can be set to the input or output mode in 1-bit units.

The pins of port 9 have the following alternate functions.

Table 4-13. Alternate Functions of Port 9

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note]
	IG4-H	IH4-H			
	GC	GF			
P90	–	18	A0	Output	Provided
P91	–	17	A1	Output	
P92	–	16	A2	Output	
P93	–	15	A3	Output	
P94	–	14	A4	Output	
P95	–	13	A5	Output	
P96	–	12	A6	Output	
P97	–	11	A7	Output	

Note Software pull-up function

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(1) Registers

(a) Port 9 register (P9)

After reset: Undefined R/W Address: FFFFF412H

	7	6	5	4	3	2	1	0
P9	P97	P96	P95	P94	P93	P92	P91	P90

P9n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 7

(b) Port 9 mode register (PM9)

After reset: FFH R/W Address: FFFFF432H

	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PM9n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 7

(c) Port 9 mode control register (PMC9)

After reset: 00H R/W Address: FFFFF452H

	7	6	5	4	3	2	1	0
PMC9	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PMC97	Specification of operating mode of P97 pin
0	I/O port
1	A7 output

PMC96	Specification of operating mode of P96 pin
0	I/O port
1	A6 output

PMC95	Specification of operating mode of P95 pin
0	I/O port
1	A5 output

PMC94	Specification of operating mode of P94 pin
0	I/O port
1	A4 output

PMC93	Specification of operating mode of P93 pin
0	I/O port
1	A3 output

PMC92	Specification of operating mode of P92 pin
0	I/O port
1	A2 output

PMC91	Specification of operating mode of P91 pin
0	I/O port
1	A1 output

PMC90	Specification of operating mode of P90 pin
0	I/O port
1	A0 output

(d) Pull-up resistor option register 9 (PU9)

After reset: 00H R/W Address: FFFFFC52H

	7	6	5	4	3	2	1	0
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90

PU9n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 7

4.3.9 Port DL

Port DL can be set to the input or output mode in 1-bit units.

The pins of port DL have the following alternate functions.

Table 4-14. Alternate Functions of Port DL

Pin Name	Pin No.		Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IG4-H	IH4-H			
	GC	GF			
PDL0	81	6	AD0	I/O	Provided
PDL1	80	5	AD1	I/O	
PDL2	79	4	AD2	I/O	
PDL3	78	3	AD3	I/O	
PDL4	77	2	AD4	I/O	
PDL5	76	1	AD5/FLMD1[Note 2]	I/O	
PDL6	75	128	AD6	I/O	
PDL7	74	127	AD7	I/O	
PDL8	73	126	AD8	I/O	
PDL9	72	125	AD9	I/O	
PDL10	71	124	AD10	I/O	
PDL11	70	123	AD11	I/O	
PDL12	69	122	AD12	I/O	
PDL13	68	121	AD13	I/O	
PDL14	67	120	AD14/TOA20/TIA20/INTP15	I/O	
PDL15	66	119	AD15/TOA21/TIA21/INTP16	I/O	

Notes 1. Software pull-up function

2. This pin is used in the flash programming mode and does not have to be manipulated by a port control register. For details, see **CHAPTER 27 FLASH MEMORY**.

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(1) Registers

(a) Port DL register (PDL)

After reset: Undefined R/W Address: PDL FFFFF004H
PDLL FFFFF004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL (PDLH[Note])	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8

	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Note To read/write bits 8 to 15 of the PDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PDLH register.

Remarks 1. The PDL register can be read or written in 16-bit units.
When the higher 8 bits of the PDL register are used as the PDLH register, and the lower 8 bits, as the PDLL register, these registers can be read or written in 8-bit or 1-bit units.

2. n = 0 to 15

(b) Port DL mode register (PMDL)

After reset: FFFFH R/W Address: PMDL FFFFF024H
PMDLL FFFFF024H, PMDLH FFFFF025H

	15	14	13	12	11	10	9	8
PMDL (PMDLH[Note])	PMDL15	PMDL14	PMDL13	PMDL12	PDAL11	PDAL10	PMDL9	PMDL8

	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Note To read/write bits 8 to 15 of the PMDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMDLH register.

Remarks 1. The PMDL register can be read or written in 16-bit units.
When the higher 8 bits of the PMDL register are used as the PMDLH register, and the lower 8 bits, as the PMDLL register, these registers can be read or written in 8-bit or 1-bit units.

2. n = 0 to 15

(c) Port DL mode control register (PMCDL)

After reset: 0000H R/W Address: PMCDL FFFFF044H
PMCDLL FFFFF044H, PMCDLH FFFFF045H

	15	14	13	12	11	10	9	8
PMCDL (PMCDLH[Note])	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8

	7	6	5	4	3	2	1	0
(PMCDLL)	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

PMCDL15	Specification of operating mode of PMCDL15 pin
0	I/O port
1	AD15 I/O/TOA21 output/TIA21 input/INTP16 input

PMCDL14	Specification of operating mode of PMCDL14 pin
0	I/O port
1	AD14 I/O/TOA20 output/TIA20 input/INTP15 input

PMCDLn	Specification of operating mode of PMCDLn pin
0	I/O port
1	ADn I/O

Note To read/write bits 8 to 15 of the PMCDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMCDLH register.

Remark n = 0 to 13

(d) Port DL function control register (PFCDL)

After reset: 0000H R/W Address: PFCDL FFFFF3A0H
PFCDLL FFFFF3A0H, PFCDLH FFFFF3A1H

	15	14	13	12	11	10	9	8
PFCDL (PFCDLH[Note])	PFCDL15	PFCDL14	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(PFCDLL)	0	0	0	0	0	0	0	0

Note To read/write bits 8 to 15 of the PFCDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PFCDLH register.

Remark For the specifications of alternate functions, see **4.3.9 (1) (f) Settings of alternate functions of port DL**.

(e) Port DL function control expansion register (PFCEDL)

After reset: 0000H R/W Address: PFCEDL FFFFF3C0H
PFCEDLL FFFFF3C0H, PFCEDLH FFFFF3C1H

	15	14	13	12	11	10	9	8
PFCEDL (PFCEDLH[Note])	PFCEDL15	PFCEDL14	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(PFCEDLL)	0	0	0	0	0	0	0	0

Note To read/write bits 8 to 15 of the PFCEDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PFCEDLH register.

Remark For the specifications of alternate functions, see **4.3.9 (1) (f) Settings of alternate functions of port DL**.

(f) Settings of alternate functions of port DL

PFCEDL15	PFCDL15	Specification of Alternate Function of PDL15 Pin
0	0	AD15 I/O
0	1	TOA21 output
1	0	TIA21 input
1	1	INTP16 input

PFCEDL14	PFCDL14	Specification of Alternate Function of PDL14 Pin
0	0	AD14 I/O
0	1	TOA20 output
1	0	TIA20 input
1	1	INTP15 input

(g) Pull-up resistor option register DL (PUDL)

After reset: 0000H R/W Address: PUDL FFFFFF44H
PUDLL FFFFFF44H, PUDLH FFFFFF45H

	15	14	13	12	11	10	9	8
PUDL (PUDLH[Note 1])	PUDL15	PUDL14	PUDL13	PUDL12	PUDL11	PUDL10	PUDL9	PUDL8

	7	6	5	4	3	2	1	0
(PUDLL)	PUDL7	PUDL6	PUDL5	PUDL4	PUDL3	PUDL2	PUDL1	PUDL0

PUDLn	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note 2]

Notes 1. To read/write bits 8 to 15 of the PUDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PUDLH register.

2. An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remarks 1. The PUDL register can be read or written in 16-bit units.
When the higher 8 bits of the PUDL register are used as the PUDLH register, and the lower 8 bits, as the PUDLL register, these registers can be read or written in 8-bit or 1-bit units.

2. n = 0 to 15

4.4 Output Data and Port Read Value for Each Setting

Table 4-15 shows the values used to select the alternate function of the respective pins, output data and port read values for each setting. In addition to the settings shown in Table 4-15, the setting of each peripheral function control register is required.

Table 4-15. Output Data and Port Read Value for Each Setting (1/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P00, P02	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TECR0, TIT00, TENC01, TIT01	1	0	0	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	TOT00, TOT01	1	0	1	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	INTP00, INTP02	1	1	0	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
P01	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TENC00	1	0	0	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	EVTT0	1	0	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	INTP01	1	1	0	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
P03 to P06	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOT20, TOT21, TOT30, TOT31	1	0	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIT20, TIT21, TIT30, TIT31	1	0	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
P03 to P06	TOT2OFF, INTP03, INTP04, TOT3OFF, INTP05, INTP06	1	1	0	0	–	Port latch	Alternate input (timer input, external interrupt input (necessary to specify valid edge))
					1		Pin level	

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (2/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P07	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB01OFF, INTP07	1	None	None	0	–	Port latch	Alternate input (timer input, external interrupt input (necessary to specify valid edge))
					1		Pin level	
	CLKOUT	1	None	1	0	Alternate output (bus output)	Port latch	
					1		Pin level	
P10 to P12	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB0T1, TOB0B1, TOB0T2	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	TIB01 to TIB03	1	0	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	TOB01 to TOB03	1	1	0	0	Alternate output 2 (timer output)	Port latch	
					1		Pin level	
P13 to P15	Output port	0	×	×	0	Port latch		
	Input port				1	–		
	TOB0B2, TOB0T3, TOB0B3	1	None	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIB00, EVTB0, TRGB0	1	None	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (3/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P16	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB00	1	0	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TOB0OFF/INTP08	1	0	1	0	–	Port latch	Alternate input (timer input, external interrupt input (necessary to specify valid edge))
					1		Pin level	
	ADTRG0/INTADT0	1	1	0	0	–	Port latch	Alternate input (A/D input, external interrupt input (necessary to specify valid edge))
					1		Pin level	
P17[Note]	Output port	None	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
P20 to P22[Note]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB1T1[Note], TOB1B1[Note], TOB1T2[Note]	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	TIB11 to TIB13[Note]	1	0	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	TOB11 to TOB13[Note]	1	1	0	0	Alternate output 2 (timer output)	Port latch	
					1		Pin level	
P23[Note], P24, P25	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB1B2[Note], TOB1T3, TOB1B3	1	None	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIB10[Note], EVTB1, TRGB1	1	None	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	

Note V850E/IH4-H only

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (4/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P26	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB10	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	TOB1OFF/INTP10	1	0	1	0	–	Port latch	Alternate input (timer input, external interrupt input (necessary to specify valid edge))
					1		Pin level	
	ADTRG1/INTADT1	1	1	0	0	Alternate output 2 (bus output)	Port latch	Alternate input (A/D input, external interrupt input (necessary to specify valid edge))
					1		Pin level	
P27	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	INTP09	1	0	0	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
	$\overline{\text{WR0}}$	1	0	1	0	Alternate output 1 (bus output)	Port latch	
					1		Pin level	
	TOA01	1	1	0	0	Alternate output 2 (timer output)	Port latch	
					1		Pin level	

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (5/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P30	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	RXDA1	1	0	0	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
	SCL	1	0	1	0	Alternate I/O (serial I/O)	Port latch	Output in master mode Input in slave mode
					1		Pin level	
	$\overline{\text{WR1}}$	1	1	0	0	Alternate output (bus output)	Port latch	
					1		Pin level	
P31	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TXDA1	1	0	0	0	Alternate output (serial output)	Port latch	
					1		Pin level	
	SDA	1	0	1	0	Alternate I/O (serial I/O)	Port latch	Output in master mode Input in slave mode
					1		Pin level	
	$\overline{\text{WAIT}}$	1	1	0	0	–	Port latch	Alternate input (bus input)
					1		Pin level	
P32	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SIF1	1	0	0	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
	RXDA2	1	0	1	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
	$\overline{\text{CS1}}$	1	1	0	0	Alternate output (bus output)	Port latch	
					1		Pin level	

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (6/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P33	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SOF1	1	None	0	0	Alternate output 1 (serial output)	Port latch	
					1		Pin level	
	TXDA2	1	None	1	0	Alternate output 2 (serial output)	Port latch	
					1		Pin level	
P34	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	$\overline{\text{SCKF1}}$	1	0	0	0	Alternate I/O (serial I/O)	Port latch	Output in master mode Input in slave mode
					1		Pin level	
	INTP11	1	0	1	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
	$\overline{\text{CS0}}$	1	1	0	0	Alternate output (bus output)	Port latch	
					1		Pin level	
P35	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SIF2	1	None	0	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
	RXDB	1	None	1	0	–	Port latch	Alternate input (serial input)
					1		Pin level	

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (7/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P36	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SOF2	1	None	0	0	Alternate output 1 (serial output)	Port latch	
					1		Pin level	
	TXDB	1	None	1	0	Alternate output 2 (serial output)	Port latch	
					1		Pin level	
P37	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	$\overline{\text{SCKF2}}$	1	0	0	0	Alternate I/O (serial I/O)	Port latch	Output in master mode Input in slave mode
					1		Pin level	
	INTP12	1	0	1	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
	ASTB	1	1	0	0	Alternate output (bus output)	Port latch	
					1		Pin level	

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (8/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P40[Note]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SIF0	1	0	0	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
	RXDA0	1	0	1	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
	TOA00	1	1	1	0	Alternate output (timer output)	Port latch	
					1		Pin level	
P41	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SOF0	1	None	0	0	Alternate output 1 (serial output)	Port latch	
					1		Pin level	
	TXDA0	1	None	1	0	Alternate output 2 (serial output)	Port latch	
					1		Pin level	

Note The P40 pin is also used for on-chip debugging. Switching between the on-chip debug function and port function (including the alternate function) can be done by using the $\overline{\text{DRST}}$ pin level. The following shows the setting method.

Port 4 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P40/SIF0/RXDA0/TOA00	DDI

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (9/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P42[Note]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	$\overline{\text{SCKF0}}$	1	0	None	0	Alternate I/O (serial I/O)	Port latch	Output in master mode Input in slave mode
					1		Pin level	
	TOA10	1	1	None	0	Alternate output (time output)	Port latch	
					1		Pin level	
P43[Note]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	INTP13	1	None	0	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
	TOA11	1	None	1	0	Alternate output (time output)	Port latch	
					1		Pin level	

Note The P42 and P43 pins are also used for on-chip debugging. Switching between the on-chip debug function and port function (including the alternate function) can be done by using the $\overline{\text{DRST}}$ pin level. The following shows the setting method.

Port 4 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P42/$\overline{\text{SCKF0}}$/TOA10	DCK
P43/INTP13/TOA11	DMS

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (10/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P44	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	INTP14	1	None	0	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
	$\overline{\text{RD}}$	1	None	1	0	Alternate output (bus output)	Port latch	
					1		Pin level	
P50, P52	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TECR1, TIT10, TENC11, TIT11	1	0	0	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	TOT10, TOT11	1	0	1	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	INTP17, INTP19	1	1	0	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
P51	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TENC10	1	0	0	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	EVTT1	1	0	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	INTP18	1	1	0	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	
	UCLK[Note]	1	1	1	0	–	Port latch	Alternate input (USB clock input)
					1		Pin level	

Note V850E/IG4-H only

Remark ×: 0 or 1

Table 4-15. Output Data and Port Read Value for Each Setting (11/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P53[Note 1]	Output port	0	×	×	0	Port latch	Port latch	Alternate input (USB clock input)
	Input port				1	–	Pin level	
	UCLK[Note 1]	1	×	×	0	–	Port latch	
					1		Pin level	
P54 to P56[Note]	Output port	None	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
P70 to P711	Input port	0	None	None	None	–	Port latch	Input-only port
	ANI20 to ANI211	1				–	Pin level	
P90 to P97[Note 1]	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	A0 to A7[Note 1]	1	None	None	0	Alternate output (bus output)	Port latch	
					1		Pin level	
PDL0 to PDL13[Note 2]	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	AD0 to AD13	1	None	None	0	Alternate output (bus output)	Port latch	
					1		Pin level	

Notes 1. V850E/IH4-H only

2. The PDL5 pin is also used in flash programming mode. This pin does not have to be manipulated by a port control register. For details, see **CHAPTER 27 FLASH MEMORY**.

Table 4-15. Output Data and Port Read Value for Each Setting (12/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
PDL14, PDL15	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1		Pin level	
	AD14, AD15	1	0	0	0	Alternate I/O (bus I/O)	Port latch	
					1		Pin level	
	TOA20, TOA21	1	0	1	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIA20, TIA21	1	1	0	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	INTP15, INTP16	1	1	1	0	–	Port latch	Alternate input (external interrupt input (necessary to specify valid edge))
					1		Pin level	

4.5 Port Register Settings When Alternate Function Is Used

The following shows the port register settings when each port is used for an alternate function. When using a port pin as an alternate-function pin, refer to the description of each pin.

Table 4-16. Settings When Pins Are Used for Alternate Functions (1/8)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bit (Register)
	Name	I/O						
P00	TECR0	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 0	PFC00 = 0	
	TIT00	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 0	PFC00 = 0	
	TOT00	Output	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 0	PFC00 = 1	
	INTP00	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 1	PFC00 = 0	INTF00 (INTF2), INTR00 (INTR2)
P01	TENC00	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 0	PFC01 = 0	
	EVTT0	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 0	PFC01 = 1	
	INTP01	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 1	PFC01 = 0	INTF01 (INTF2), INTR01 (INTR2)
P02	TENC01	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 0	PFC02 = 0	
	TIT01	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 0	PFC02 = 0	
	TOT01	Output	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 0	PFC02 = 1	
	INTP02	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 1	PFC02 = 0	INTF02 (INTF2), INTR02 (INTR2)
P03	TOT20	Output	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 0	PFC03 = 0	
	TIT20	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 0	PFC03 = 1	
	TOT2OFF	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 1	PFC03 = 0	
	INTP03	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 1	PFC03 = 0	INTF03 (INTF0), INTR03 (INTR0)
P04	TOT21	Output	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 0	
	TIT21	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 1	
	INTP04	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 1	PFC04 = 0	INTF04 (INTF0), INTR04 (INTR0)
P05	TOT30	Output	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 0	PFC05 = 0	
	TIT30	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 0	PFC05 = 1	
	TOT3OFF	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 1	PFC05 = 0	
	INTP05	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 1	PFC05 = 0	INTF05 (INTF0), INTR05 (INTR0)

Table 4-16. Settings When Pins Are Used for Alternate Functions (2/8)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bit (Register)
	Name	I/O						
P06	TOT31	Output	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	PFCE06 = 0	PFC06 = 0	
	TIT31	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	PFCE06 = 0	PFC06 = 1	
	INTP06	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	PFCE06 = 1	PFC06 = 0	INTF06 (INTF0), INTR06 (INTR0)
P07	TOB01OFF	Input	P07 = Setting not required	PM07 = Setting not required	PMC07 = 1	–	PFC07 = 0	
	INTP07	Input	P07 = Setting not required	PM07 = Setting not required	PMC07 = 1	–	PFC07 = 0	INTF07 (INTF0), INTR07 (INTR0)
	CLKOUT	Output	P07 = Setting not required	PM07 = Setting not required	PMC07 = 1	–	PFC07 = 1	
P10	TOB0T1	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 0	PFC10 = 0	
	TIB01	Input	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 0	PFC10 = 1	
	TOB01	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 1	PFC10 = 0	
P11	TOB0B1	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 0	PFC11 = 0	
	TIB02	Input	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 0	PFC11 = 1	
	TOB02	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 1	PFC11 = 0	
P12	TOB0T2	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 0	PFC12 = 0	
	TIB03	Input	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 0	PFC12 = 1	
	TOB03	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 1	PFC12 = 0	
P13	TOB0B2	Output	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	–	PFC13 = 0	
	TIB00	Input	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	–	PFC13 = 1	
P14	TOB0T3	Output	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	–	PFC14 = 0	
	EVTB0	Input	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	–	PFC14 = 1	
P15	TOB0B3	Output	P15 = Setting not required	PM15 = Setting not required	PMC15 = 1	–	PFC15 = 0	
	TRGB0	Input	P15 = Setting not required	PM15 = Setting not required	PMC15 = 1	–	PFC15 = 1	

Table 4-16. Settings When Pins Are Used for Alternate Functions (3/8)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bit (Register)
	Name	I/O						
P16	TOB00	Output	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 0	PFC16 = 0	
	TOB0OFF	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 0	PFC16 = 1	
	INTP08	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 0	PFC16 = 1	INTF08 (INTF0), INTR08 (INTR0)
	ADTRG0	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 1	PFC16 = 0	
	INTADA0	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 1	PFC16 = 0	ADTF0 (ADTF), ADTR0 (ADTR)
P17[Note]	–	–	P17 = Setting not required	PM17 = Setting not required	–	–	–	
P20[Note]	TOB1TI[Note]	Output	P20 = Setting not required	PM20 = Setting not required	PMC20 = 1	PFCE20 = 0	PFC20 = 0	
	TIB11[Note]	Input	P20 = Setting not required	PM20 = Setting not required	PMC20 = 1	PFCE20 = 0	PFC20 = 1	
	TOB11[Note]	Output	P20 = Setting not required	PM20 = Setting not required	PMC20 = 1	PFCE20 = 1	PFC20 = 0	
P21[Note]	TOB1B1[Note]	Output	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 0	PFC21 = 0	
	TIB12[Note]	Input	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 0	PFC21 = 1	
	TOB12[Note]	Output	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 1	PFC21 = 0	
P22[Note]	TOB1T2[Note]	Output	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 0	PFC22 = 0	
	TIB13[Note]	Input	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 0	PFC22 = 1	
	TOB13[Note]	Output	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 1	PFC22 = 0	
P23[Note]	TOB1B2[Note]	Output	P23 = Setting not required	PM23 = Setting not required	PMC23 = 1	–	PFC23 = 0	
	TIB10[Note]	Input	P23 = Setting not required	PM23 = Setting not required	PMC23 = 1	–	PFC23 = 1	
P24	TOB1T3	Output	P24 = Setting not required	PM24 = Setting not required	PMC24 = 1	–	PFC24 = 0	
	EVTB1	Input	P24 = Setting not required	PM24 = Setting not required	PMC24 = 1	–	PFC24 = 1	
P25	TOB1B3	Output	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	–	PFC25 = 0	
	TRGB1	Input	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	–	PFC25 = 1	

Note V850E/IH4-H only

Table 4-16. Settings When Pins Are Used for Alternate Functions (4/8)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bit (Register)
	Name	I/O						
P26	TOB10	Output	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 0	PFC26 = 0	
	TOB1OFF	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 0	PFC26 = 1	
	INTP10	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 0	PFC26 = 1	INTF10 (INTF0), INTR10 (INTR0)
	ADTRG1	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 1	PFC26 = 0	
	INTADT1	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 1	PFC26 = 0	ADTF1 (ADTF), ADTR1 (ADTR)
P27	INTP09	Input	P27 = Setting not required	PM27 = Setting not required	PMC27 = 1	PFCE27 = 0	PFC27 = 0	INTF09 (INTF0), INTR09 (INTR0)
	$\overline{\text{WR0}}$	Output	P27 = Setting not required	PM27 = Setting not required	PMC27 = 1	PFCE27 = 0	PFC27 = 1	
	TOA01	Output	P27 = Setting not required	PM27 = Setting not required	PMC27 = 1	PFCE27 = 1	PFC27 = 0	
P30	RXDA1	Input	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFCE30 = 0	PFC30 = 0	
	SCL	I/O	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFCE30 = 0	PFC30 = 1	PF30 (PF3) = 1
	$\overline{\text{WR1}}$	Output	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFCE30 = 1	PFC30 = 0	
P31	TXDA1	Output	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	PFCE31 = 0	PFC31 = 0	
	SDA	I/O	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	PFCE31 = 0	PFC31 = 1	PF31 (PF3) = 1
	$\overline{\text{WAIT}}$	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	PFCE31 = 1	PFC31 = 0	
P32	SIF1	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 0	
	RXDA2	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 1	
	$\overline{\text{CS1}}$	Output	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 0	
P33	SOF1	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 0	
	TXDA2	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 1	
P34	$\overline{\text{SCKF1}}$	I/O	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 0	PFC34 = 0	
	INTP11	Input	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 0	PFC34 = 1	INTF11 (INTF1), INTR11 (INTR1)
	$\overline{\text{CS0}}$	Output	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 1	PFC34 = 0	
P35	SIF2	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	PFC35 = 0	
	RXDB	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	PFC35 = 1	

Table 4-16. Settings When Pins Are Used for Alternate Functions (5/8)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bit (Register)
	Name	I/O						
P36	SOF2	Output	P36 = Setting not required	PM36 = Setting not required	PMC36 = 1	–	PFC36 = 0	
	TXDB	Output	P36 = Setting not required	PM36 = Setting not required	PMC36 = 1	–	PFC36 = 1	
P37	$\overline{\text{SCKF2}}$	I/O	P37 = Setting not required	PM37 = Setting not required	PMC37 = 1	PMCE37 = 0	PFC37 = 0	
	INTP12	Input	P37 = Setting not required	PM37 = Setting not required	PMC37 = 1	PMCE37 = 0	PFC37 = 1	INTF12 (INTF1), INTR12 (INTR1)
	ASTB	Output	P37 = Setting not required	PM37 = Setting not required	PMC37 = 1	PMCE37 = 1	PFC37 = 0	
P40	SIF0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	PMCE40 = 0	PFC40 = 0	
	RXDA0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	PMCE40 = 0	PFC40 = 1	
	DDI[Note]	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = Setting not required	PFCE40 = Setting not required	PFC40 = Setting not required	
	TOA00	Output	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	PFCE40 = 1	PFC40 = 1	
P41	SOF0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	PFC41 = 0	
	TXDA0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	PFC41 = 1	
P42	$\overline{\text{SCKF0}}$	I/O	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	PFCE42 = 0	–	
	DCK[Note]	Input	P42 = Setting not required	PM42 = Setting not required	PMC42 = Setting not required	PFCE42 = Setting not required	–	
	TOA10	Output	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	PFCE42 = 1	–	
P43	INTP13	Input	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	–	PFC43 = 0	INTF13 (INTF1), INTR13 (INTR1)
	DMS[Note]	Input	P43 = Setting not required	PM43 = Setting not required	PMC43 = Setting not required	–	PFC43 = Setting not required	
	TOA11	Output	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	–	PFC43 = 1	

Note The P40, P42, and P43 pins are also used for on-chip debugging. Switching between the on-chip debug function and port function (including the alternate function) can be done by using the $\overline{\text{DRST}}$ pin level. The following shows the setting method.

Port 4 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P40/SIF0/RXDA0/TOA00	DDI
P42/$\overline{\text{SCKF0}}$/TOA10	DCK
P43/INTP13/TOA11	DMS

Table 4-16. Settings When Pins Are Used for Alternate Functions (6/8)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bit (Register)
	Name	I/O						
P44	INTP14	Input	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	–	PFC44 = 0	INTF14 (INTF1), INTR14 (INTR1)
	$\overline{\text{RD}}$	Output	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	–	PFC44 = 1	
P50	TECR1	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 0	
	TIT10	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 0	
	TOT10	Output	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 1	
	INTP17	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 0	INTF17 (INTF3), INTR17 (INTR3)
P51	TENC10	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 0	
	EVTT1	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 1	
	INTP18	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 1	PFC51 = 0	INTF18 (INTF3), INTR18 (INTR3)
	UCLK[Note 1]	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 1	PFC51 = 1	
P52	TENC11	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 0	
	TIT11	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 0	
	TOT11	Output	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 1	
	INTP19	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 1	PFC52 = 0	INTF19 (INTF3), INTR19 (INTR3)
P53[Note 2]	UCLK[Note 2]	Input	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	–	–	
P54[Note 2]	–	–	P54 = Setting not required	PM54 = Setting not required	–	–	–	
P55[Note 2]	–	–	P55 = Setting not required	PM55 = Setting not required	–	–	–	
P56[Note 2]	–	–	P56 = Setting not required	PM56 = Setting not required	–	–	–	
P70	ANI20	Input	P70 = Setting not required	–	PMC70 = 1	–	–	
P71	ANI21	Input	P71 = Setting not required	–	PMC71 = 1	–	–	
P72	ANI22	Input	P72 = Setting not required	–	PMC72 = 1	–	–	
P73	ANI23	Input	P73 = Setting not required	–	PMC73 = 1	–	–	

Notes 1. V850E/IG4-H only

2. V850E/IH4-H only

Table 4-16. Settings When Pins Are Used for Alternate Functions (7/8)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bit (Register)
	Name	I/O						
P74	ANI24	Input	P74 = Setting not required	–	PMC74 = 1	–	–	
P75	ANI25	Input	P75 = Setting not required	–	PMC75 = 1	–	–	
P76	ANI26	Input	P76 = Setting not required	–	PMC76 = 1	–	–	
P77	ANI27	Input	P77 = Setting not required	–	PMC77 = 1	–	–	
P78	ANI28	Input	P78 = Setting not required	–	PMC78 = 1	–	–	
P79	ANI29	Input	P79 = Setting not required	–	PMC79 = 1	–	–	
P710	ANI210	Input	P710 = Setting not required	–	PMC710 = 1	–	–	
P711	ANI211	Input	P711 = Setting not required	–	PMC711 = 1	–	–	
P90[Note]	A0[Note]	Output	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	–	–	
P91[Note]	A1[Note]	Output	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	–	–	
P92[Note]	A2[Note]	Output	P92 = Setting not required	PM92 = Setting not required	PMC92 = 1	–	–	
P93[Note]	A3[Note]	Output	P93 = Setting not required	PM93 = Setting not required	PMC93 = 1	–	–	
P94[Note]	A4[Note]	Output	P94 = Setting not required	PM94 = Setting not required	PMC94 = 1	–	–	
P95[Note]	A5[Note]	Output	P95 = Setting not required	PM95 = Setting not required	PMC95 = 1	–	–	
P96[Note]	A6[Note]	Output	P96 = Setting not required	PM96 = Setting not required	PMC96 = 1	–	–	
P97[Note]	A7[Note]	Output	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	–	–	
PDL0	AD0	I/O	PDL0 = Setting not required	PMDL0 = Setting not required	PMCDL0 = 1	–	–	
PDL1	AD1	I/O	PDL1 = Setting not required	PMDL1 = Setting not required	PMCDL1 = 1	–	–	
PDL2	AD2	I/O	PDL2 = Setting not required	PMDL2 = Setting not required	PMCDL2 = 1	–	–	
PDL3	AD3	I/O	PDL3 = Setting not required	PMDL3 = Setting not required	PMCDL3 = 1	–	–	
PDL4	AD4	I/O	PDL4 = Setting not required	PMDL4 = Setting not required	PMCDL4 = 1	–	–	

Note V850E/IH4-H only

Table 4-16. Settings When Pins Are Used for Alternate Functions (8/8)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bit (Register)
	Name	I/O						
PDL5	AD5	I/O	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = 1	–	–	
	FLMD1[Note]	Input	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = Setting not required	–	–	
PDL6	AD6	I/O	PDL6 = Setting not required	PMDL6 = Setting not required	PMCDL6 = 1	–	–	
PDL7	AD7	I/O	PDL7 = Setting not required	PMDL7 = Setting not required	PMCDL7 = 1	–	–	
PDL8	AD8	I/O	PDL8 = Setting not required	PMDL8 = Setting not required	PMCDL8 = 1	–	–	
PDL9	AD9	I/O	PDL9 = Setting not required	PMDL9 = Setting not required	PMCDL9 = 1	–	–	
PDL10	AD10	I/O	PDL10 = Setting not required	PMDL10 = Setting not required	PMCDL10 = 1	–	–	
PDL11	AD11	I/O	PDL11 = Setting not required	PMDL11 = Setting not required	PMCDL11 = 1	–	–	
PDL12	AD12	I/O	PDL12 = Setting not required	PMDL12 = Setting not required	PMCDL12 = 1	–	–	
PDL13	AD13	I/O	PDL13 = Setting not required	PMDL13 = Setting not required	PMCDL13 = 1	–	–	
PDL14	AD14	I/O	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	PFCEDL14 = 0	PFCDL14 = 0	
	TOA20	Output	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	PFCEDL14 = 0	PFCDL14 = 1	
	TIA20	Input	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	PFCEDL14 = 1	PFCDL14 = 0	
	INTP15	Input	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	PFCEDL14 = 1	PFCDL14 = 1	INTF15 (INTF1), INTR15 (INTR1)
PDL15	AD15	I/O	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	PFCEDL15 = 0	PFCDL15 = 0	
	TOA21	Output	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	PFCEDL15 = 0	PFCDL15 = 1	
	TIA21	Input	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	PFCEDL15 = 1	PFCDL15 = 0	
	INTP16	Input	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	PFCEDL15 = 1	PFCDL15 = 1	INTF16 (INTF1), INTR16 (INTR1)

Note The PDL5 pin is also used in flash programming mode. This pin does not have to be manipulated by a port control register. For details, see **CHAPTER 27 FLASH MEMORY**.

4.6 Noise Eliminator

A timing controller used to wait until the noise is eliminated is provided for the following pins. Input signals that change within the noise elimination time are not internally acknowledged.

Cautions 1. The maskable interrupt pins can be used to release the standby mode. For details, see CHAPTER 20 STANDBY FUNCTION.

2. The digital filter uses clock sampling and therefore cannot acknowledge an input signal when the peripheral clock (f_{XX}) is stopped (STOP or IDLE mode).

3. The noise eliminator is valid only in the alternate-function mode.

Table 4-17. Noise Eliminator (1/2)

Target Pin		Filter Type	Noise Elimination Width	Sampling Clock
$\overline{\text{RESET}}$		Analog filter	Several 10 ns	–
$\overline{\text{DRST}}$				
FLMD0				
P00/TECR0/TIT00/TOT00/INTP00	TECR0/TIT00	Digital filter	2, 3 clocks	fxx/2, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	INTP00	Analog filter	Several 10 ns	–
		Digital filter	2, 3 clocks	fxx/4, fxx/8, fxx/16, fxx/64, fxx/256, fxx/1024 selectable
P01/TENC00/EVTT0/INTP01	TENC00	Digital filter	2, 3 clocks	fxx/2, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	EVTT0			
	INTP01	Analog filter	Several 10 ns	–
		Digital filter	2, 3 clocks	fxx/4, fxx/8, fxx/16, fxx/64, fxx/256, fxx/1024 selectable
P02/TENC01/TIT01/TOT01/INTP02	TENC0/TIT01	Digital filter	2, 3 clocks	fxx/2, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	INTP02	Analog filter	Several 10 ns	–
		Digital filter	2, 3 clocks	fxx/4, fxx/8, fxx/16, fxx/64, fxx/256, fxx/1024 selectable
P03/TOT20/TIT20/TOT2OFF/INTP03	TIT20	Digital filter	2, 3 clocks	fxx/2, fxx/8 selectable
	TOT2OFF	Analog filter	Several 10 ns	–
	INTP03			
P04/TOT21/TIT21/INTP04	TIT21	Digital filter	2, 3 clocks	fxx/2, fxx/8 selectable
	INTP04	Analog filter	Several 10 ns	–
P05/TOT30/TIT30/TOT3OFF/INTP05	TIT30	Digital filter	2, 3 clocks	fxx/2, fxx/8 selectable
	TOT3OFF	Analog filter	Several 10 ns	–
	INTP05			
P06/TOT31/TIT31/INTP06	TIT31	Digital filter	2, 3 clocks	fxx/2, fxx/8 selectable
	INTP06	Analog filter	Several 10 ns	–
P07/TOB01OFF/INTP07/CLKOUT	TOB01OFF			
	INTP07			
P10/TOB0T1/TIB01/TOB01	TIB01	Digital filter	3 clocks	fxx/8
P11/TOB0B1/TIB02/TOB02	TIB02			
P12/TOB0T2/TIB03/TOB03	TIB03			
P13/TOB0B2/TIB00	TIB00			
P14/TOB0T3/EVTB0	EVTB0			
P15/TOB0B3/TRGB0	TRGB0			
P16/TOB00/TOB0OFF/INTP08/ADTRG0/ INTADT0	TOB0OFF	Analog filter	Several 10 ns	–
	INTP08			
	ADTRG0			
	INTADT0			

Table 4-17. Noise Eliminator (2/2)

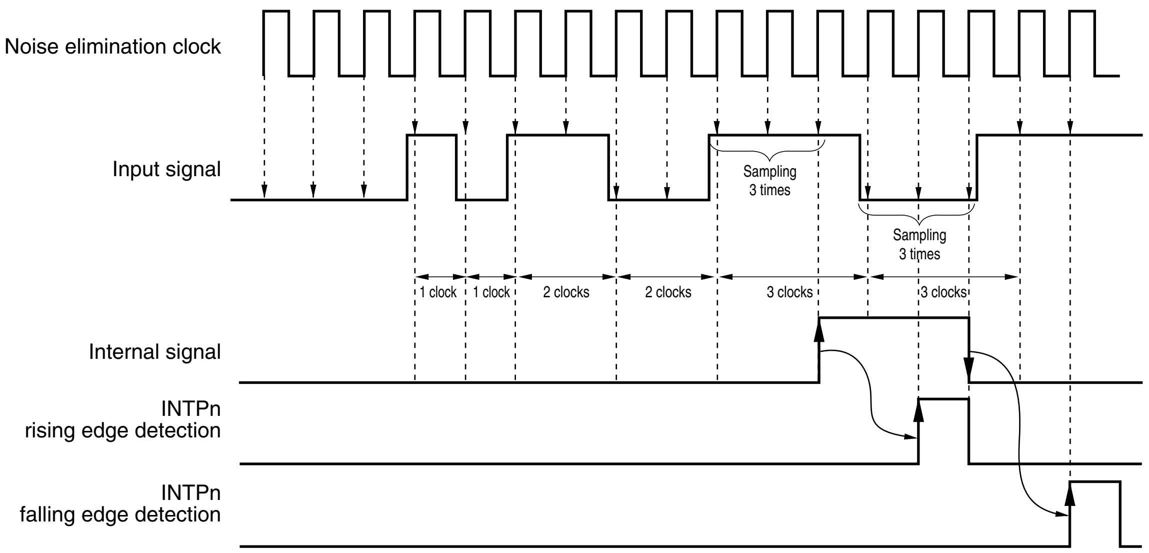
Target Pin		Filter Type	Noise Elimination Width	Sampling Clock
P20[Note 1]/TOB1T1[Note 1]/TIB11[Note 1]/TOB11[Note 1]	TIB11[Note 1]	Digital filter	3 clocks	fxx/8
P21[Note 1]/TOB1B1[Note 1]/TIB12[Note 1]/TOB12[Note 1]	TIB12[Note 1]			
P22[Note 1]/TOB1T2[Note 1]/TIB13[Note 1]/TOB13[Note 1]	TIB13[Note 1]			
P23[Note 1]/TOB1B2[Note 1]/TIB10[Note 1]	TIB10[Note 1]			
P24/TOB1T3/EVTB1	EVTB1			
P25/TOB1B3/TRGB1	TRGB1			
P26/TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1	TOB1OFF	Analog filter	Several 10 ns	–
	INTP10			
	ADTRG1			
	INTADT1			
P27/INTP09/$\overline{\text{WR0}}$/TOA01	INTP09			
P34/$\overline{\text{SCKF1}}$/INTP11/$\overline{\text{CS0}}$	INTP11			
P37/$\overline{\text{SCKF2}}$/INTP12/ASTB	INTP12			
P43/INTP13/DMS/TOA11	INTP13			
P44/INTP14/$\overline{\text{RD}}$	INTP14			
P50/TECR1/TIT10/TOT10/INTP17	TECR1	Digital filter	2, 3 clocks	fxx/2, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	TIT10			
	INTP17	Analog filter	Several 10 ns	–
		Digital filter	2, 3 clocks	fxx/4, fxx/8, fxx/16, fxx/64, fxx/256, fxx/1024 selectable
P51/TENC10/EVTT1/INTP18/UCLK[Note 2]	TENC10	Digital filter	2, 3 clocks	fxx2, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	EVTT1			
	INTP18	Analog filter	Several 10 ns	–
		Digital filter	2, 3 clocks	fxx/4, fxx/8, fxx/16, fxx/64, fxx/256, fxx/1024 selectable
P52/TENC11/TIT11/TOT11/INTP19	TENC11	Digital filter	2, 3 clocks	fxx2, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	TIT11			
	INTP19	Analog filter	Several 10 ns	–
		Digital filter	2, 3 clocks	fxx/4, fxx/8, fxx/16, fxx/64, fxx/256, fxx/1024 selectable
PDL14/AD14/TOA20/TIA20/INTP15	TIA20	Digital filter	2, 3 clocks	fxx/2, fxx/8 selectable
	INTP15	Analog filter	Several 10 ns	–
PDL15/AD15/TOA21/TIA21/INTP16	TIA21	Digital filter	2, 3 clocks	fxx/2, fxx/8 selectable
	INTP16	Analog filter	Several 10 ns	–

Notes 1. V850E/IH4-H only

2. V850E/IG4-H only

An example of timing of noise elimination by digital filtering for INTP00 to INTP02, INTP17 to INTP19, timer AA input pin, and timer T input pin is shown below.

Figure 4-4. Example of Noise Elimination Timing

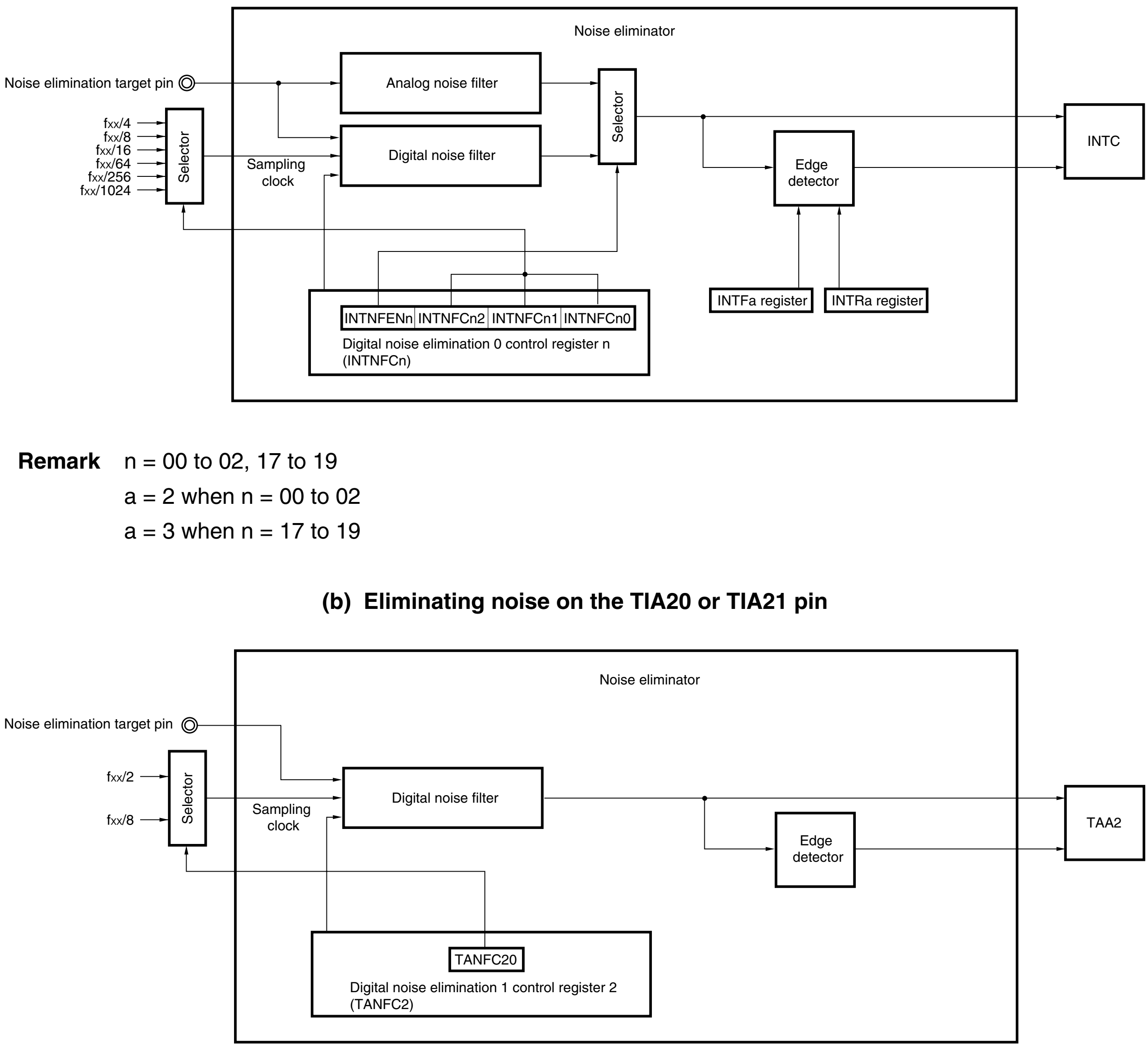


Caution **If there are two or fewer noise elimination clocks while the input signal subject to filtering is high level (or low level), the input signal is eliminated as noise. If it is sampled three times or more, the edge is detected as a valid input.**

Remark n = 00 to 02, 17 to 19

The components of the noise eliminator are shown below.

Figure 4-5. Components of Noise Eliminator (1/2)



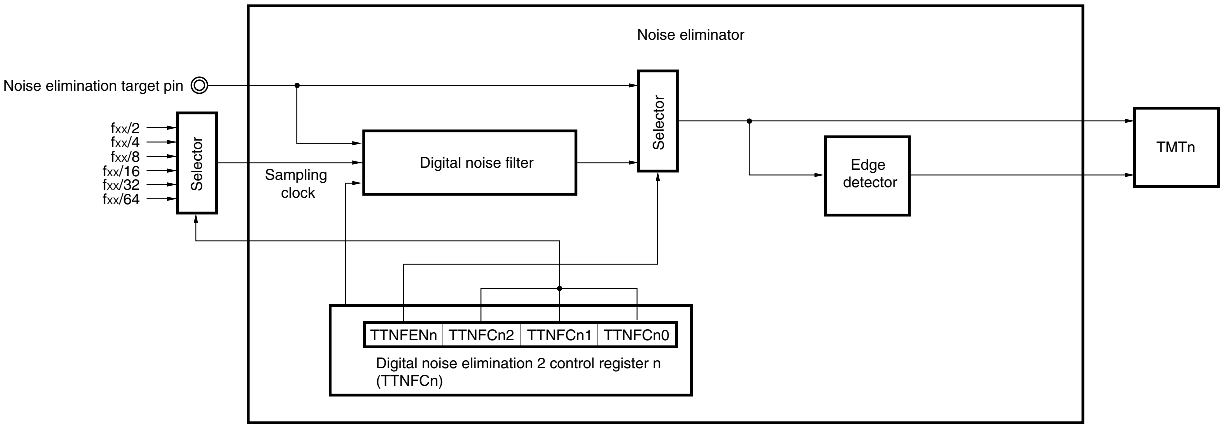
(a) Eliminating noise on the INTP00 to INTP02, or INTP17 to INTP19 pins

Remark n = 00 to 02, 17 to 19
a = 2 when n = 00 to 02
a = 3 when n = 17 to 19

(b) Eliminating noise on the TIA20 or TIA21 pin

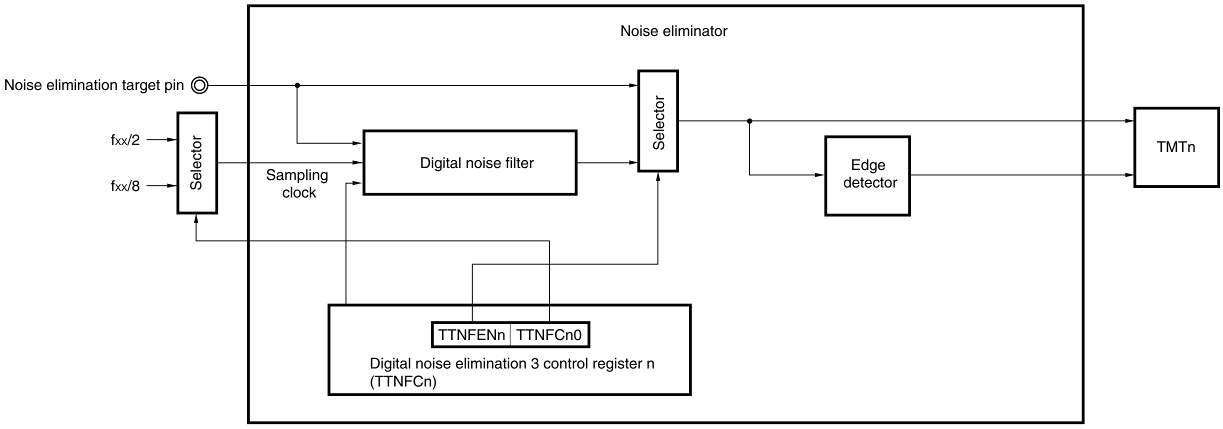
Figure 4-5. Components of Noise Eliminator (2/2)

(c) Eliminating noise on the TIT00, TIT01, EVTT0, TENC00, TENC01, TECR0, TIT10, TIT11, EVTT1, TENC10, TENC11, or TECR1 pin



Remark n = 0, 1

(d) Eliminating noise on the TIT20, TIT21, TIT30, or TIT31 pin



Remark n = 2, 3

(1) Digital noise elimination 0 control register n (INTNFCn)

The INTNFCn register is used to select the sampling clock that is used to eliminate digital noise on the INTPn pin. If the same level is not detected on this pin three times in sequence using the clock selected by the INTNFCn register, the signal is eliminated as noise.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

Cautions 1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.

3. Noise is not eliminated if the pin is used as a normal input port pin.

After reset: 00H R/W Address: INTNFC00 FFFFF310H, INTNFC01 FFFFF312H,
INTNFC02 FFFFF314H, INTNFC17 FFFFF318H,
INTNFC18 FFFFF31AH, INTNFC19 FFFFF31CH

	7	6	5	4	3	2	1	0
INTNFCn	INTNFENn	0	0	0	0	INTNFCn2	INTNFCn1	INTNFCn0

(n = 00 to 02, 17 to 19)

INTNFENn	Setting of digital noise elimination
0	Enables analog noise elimination
1	Enables digital noise elimination

INTNFCn2	INTNFCn1	INTNFCn0	Sampling clock selection
0	0	0	f_{xx}/4
0	0	1	f_{xx}/8
0	1	0	f_{xx}/16
0	1	1	f_{xx}/64
1	0	0	f_{xx}/256
1	0	1	f_{xx}/1024
Other than above			Setting prohibited

(2) Digital noise elimination 1 control register 2 (TANFC2)

The TANFC2 register is used to select the sampling clock that is used to eliminate digital noise on the TIA20 or TIA21 pin. If the same level is not detected on these pins three times in sequence using the clock selected by the TANFC2 register, the signal is eliminated as noise.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

Cautions 1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.

3. Noise is not eliminated if the pin is used as a normal input port pin.

4. The noise elimination function starts operating when the TAA2CTL0.TAA2CE bit is set to 1 (enabling count operations).

After reset: 00H R/W Address: FFFFFB40H

	7	6	5	4	3	2	1	0
TANFC2	0	0	0	0	0	0	0	TANFC20

TANFC20	Sampling clock selection
0	$f_{xx}/2$
1	$f_{xx}/8$

(3) Digital noise elimination 2 control register n (TTNFCn)

The TTNFCn register is used to select the sampling clock that is used to eliminate digital noise on the TITn0, TITn1, EVTTn, TENCn0, TENCn1, or TECRn pin. If the same level is not detected on these pins three times in sequence using the clock selected by the TTNFCn register, the signal is eliminated as noise.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

Cautions 1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.

3. Noise is not eliminated if the pin is used as a normal input port pin.

4. The noise elimination function starts operating when the TTnCTL0.TTnCE bit is set to 1 (enabling count operations).

After reset: 00H R/W Address: TTNFC0 FFFFF5A0H, TTNFC1 FFFFF5A2H

	7	6	5	4	3	2	1	0
TTNFCn (n = 0, 1)	TTNFENn	0	0	0	0	TTNFCn2	TTNFCn1	TTNFCn0

TTNFENn	Setting of digital noise elimination
0	Disables digital noise elimination
1	Enables digital noise elimination

TTNFCn2	TTNFCn1	TTNFCn0	Sampling clock selection
0	0	0	fxx/2
0	0	1	fxx/4
0	1	0	fxx/8
0	1	1	fxx/16
1	0	0	fxx/32
1	0	1	fxx/64
Other than above			Setting prohibited

(4) Digital noise elimination 3 control register n (TTNFCn)

The TTNFCn register is used to select the sampling clock that is used to eliminate digital noise on the TITn0 or TITn1 pin. If the same level is not detected on these pins three times in a row by using the clock selected by the TTNFCn register, the signal is eliminated as noise.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

Cautions 1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. To actually detect the signal as a valid edge, the same signal level must be input for a duration of 3 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by applying a filter to the input pin.

3. Noise is not eliminated if the pin is used as a normal input port pin.

4. Noise elimination starts when the TTnCTL0.TTnCE bit is set to 1 (enabling counting).

After reset: 00H R/W Address: TTNFC2 FFFFF7A0H, TTNFC3 FFFFF7A2H

	7	6	5	4	3	2	1	0
TTNFCn (n = 2, 3)	TTNFENn	0	0	0	0	0	0	TTNFCn0

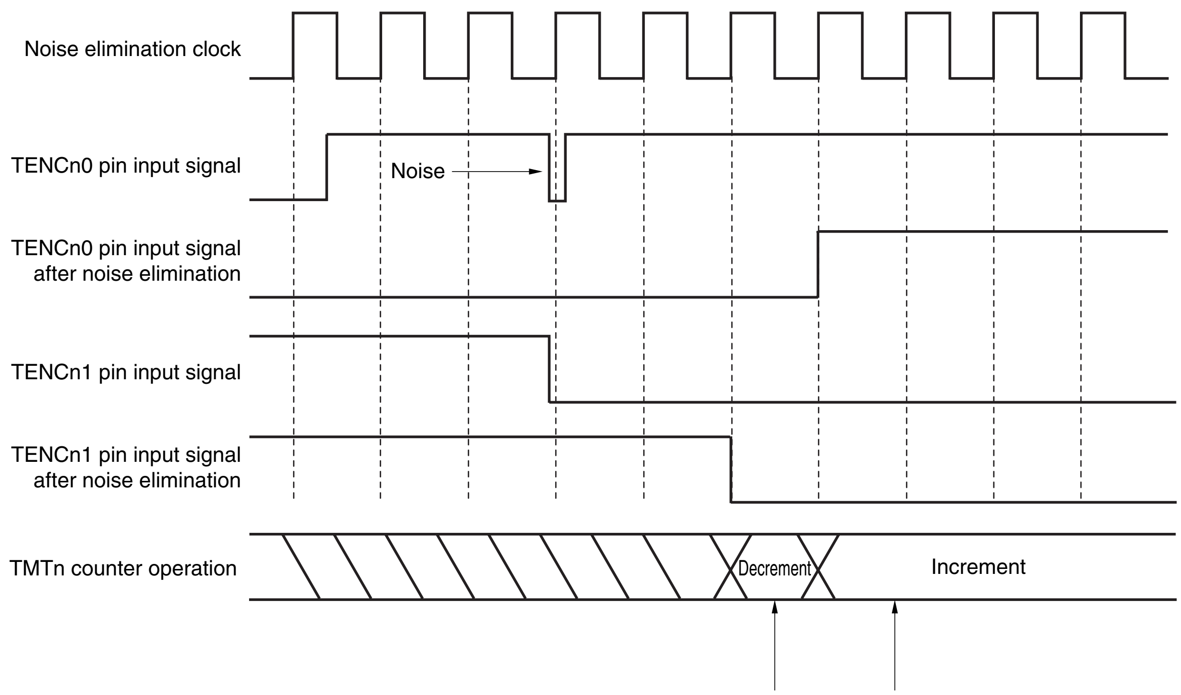
TTNFENn	Setting of digital noise elimination
0	Disables digital noise elimination
1	Enables digital noise elimination

TTNFCn0	Sampling clock selection
0	fxx/2
1	fxx/8

(a) Cautions on eliminating noise on the TENCn0 and TENCn1 pins

When eliminating noise on the TENCn0 and TENCn1 pins (TTNFCn.TTNFENn bit = 1), the following malfunction might occur.

Figure 4-6. Malfunction in Eliminating Noise on TENCn0 and TENCn1 Pins



Because of the setting of the TTnUDS1 and TTnUDS0 bits and the phase of the TENCn0 and TENCn1 pins, the TMTn operation of incrementing the counter and then decrementing the counter is originally expected.

Remark The above figure shows the timing when detecting the valid edge on both edges is specified for the TENCn0 and TENCn1 pins (TTnIOC3.TTnEIS1 and TTnIOC3.TTnEIS0 bits are set to 11) and incrementing the counter upon detection of the valid edge of the TENCn0 pin input signal and decrementing the counter upon detection of the valid edge of the TENCn1 pin input signal are specified (TTnCTL2.TTnUDS1 and TTnCTL2.TTnUDS0 bits are set to 01).

4.7 Cautions

4.7.1 Cautions on setting port pins

(1) Set the registers of a port in the following sequence.

<1> Set the PFCn and PFCEn registers.
<2> Set the PMCn register.
<3> Set the INTFn and INTRn registers.

If the PMCn register is set before setting the PFCn and PFCEn registers, an unexpected peripheral function may be selected while the PFCn and PFCEn registers are being set.

(2) An on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode.
For V850E/IG4-H, an on-chip pull-up resistor can also be connected to the TOT21, TOT31, TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, TOB1T3, and TOB1B3 pins, which function as output pins in the alternate-function mode, when these pins go into a high-impedance state due to a signal input to the TOT2OFF, TOT3OFF, TOB0OFF, TOB01OFF, or TOB1OFF pin or software processing.
For V850E/IH4-H, an on-chip pull-up resistor can also be connected to the TOT21, TOT31, TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, TOB1T1 to TOB1T3, and TOB1B1 to TOB1B3 pins, which function as output pins in the alternate-function mode, when these pins go into a high-impedance state due to a signal input to the TOT2OFF, TOT3OFF, TOB0OFF, TOB01OFF, or TOB1OFF pin or software processing.
Set the on-chip pull-up resistor in the following sequence.

<1> Set the PUn register.
<2> Set the PMCn register.
<3> Set the PMn register.

(3) Set the N-ch open-drain in the following sequence.
• Used in port mode
<1> Set the PMCn register.
<2> Set the PFn register.

• Used as output pin in alternate-function mode of I^{2}C
<1> Set the PFCn and PFCEn registers.
<2> Set the PFn register.
<3> Set the PMCn register.

4.7.2 Cautions on bit manipulation instruction for port n register (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the value of the output latch of an input port that is not subject to manipulation may be written in addition to the targeted bit.

Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

<Example> When P00 pin is an output port, P01 to P07 pins are input ports (all pin statuses are high level), and the value of the port latch is 00H, if the output of P00 pin is changed from low level to high level via a bit manipulation instruction, the value of the port latch is FFH.

Explanation: The target bits of writing to and reading from the Pn register of a port whose PMnm bit is 1 are in the output latch status and pin status, respectively.

A bit manipulation instruction is executed in the following order in the V850E/IG4-H and V850E/IH4-H.

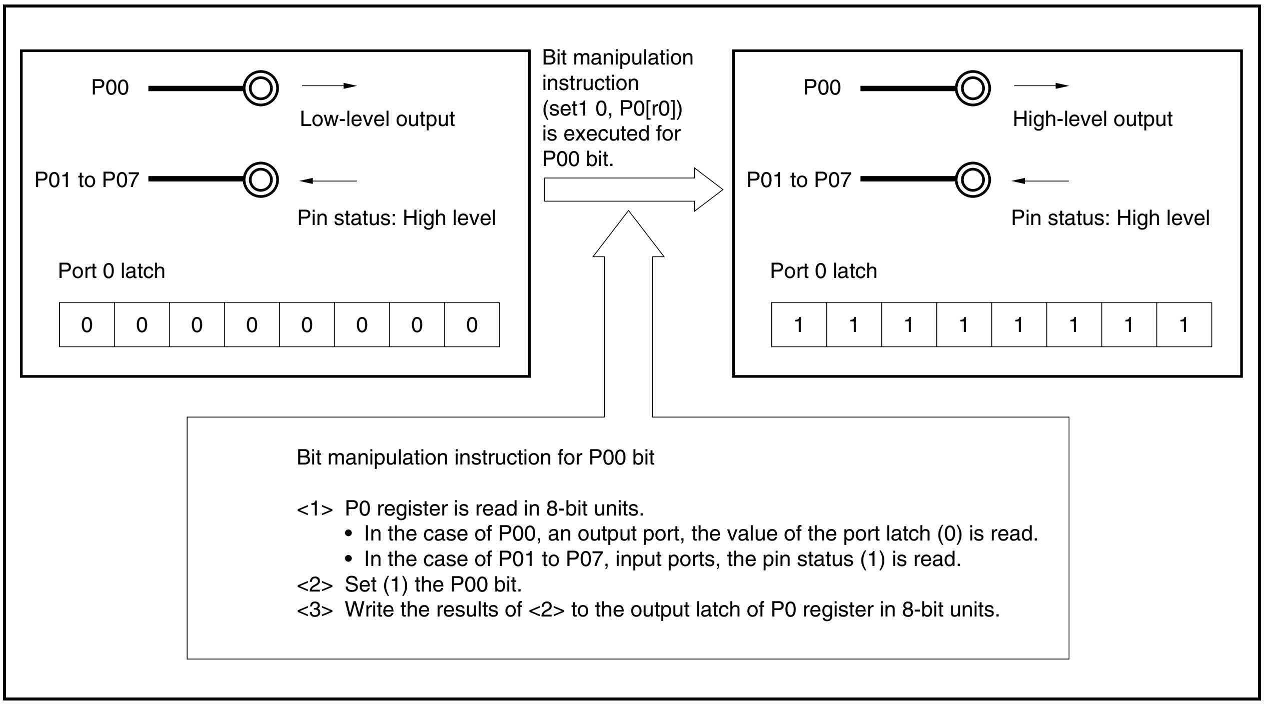
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the value of the output latch (0) of P00 pin, which is an output port, is read, while the pin statuses of P01 to P07 pins, which are input ports, are read. If the pin statuses of P01 to P07 pins are high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

Figure 4-7. Bit Manipulation Instruction (P00 Pin)



Bit manipulation instruction for P00 bit

<1> P0 register is read in 8-bit units.
- In the case of P00, an output port, the value of the port latch (0) is read.
- In the case of P01 to P07, input ports, the pin status (1) is read.

<2> Set (1) the P00 bit.
<3> Write the results of <2> to the output latch of P0 register in 8-bit units.

CHAPTER 5 CLOCK GENERATOR

5.1 Overview

The features of clock generator are as follows.

○ Oscillator

- In PLL mode: f_X = 10 to 12.5 MHz (f_{XX} = 80 to 100 MHz)
- In clock-through mode: f_X = 10 to 12.5 MHz (f_{XX} = 10 to 12.5 MHz)

○ Multiply (×8 fixed) function by PLL (Phase Locked Loop)

- Clock-through mode/PLL mode selectable

○ Internal system clock generation

- 4 steps (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$)

○ Peripheral clock generation

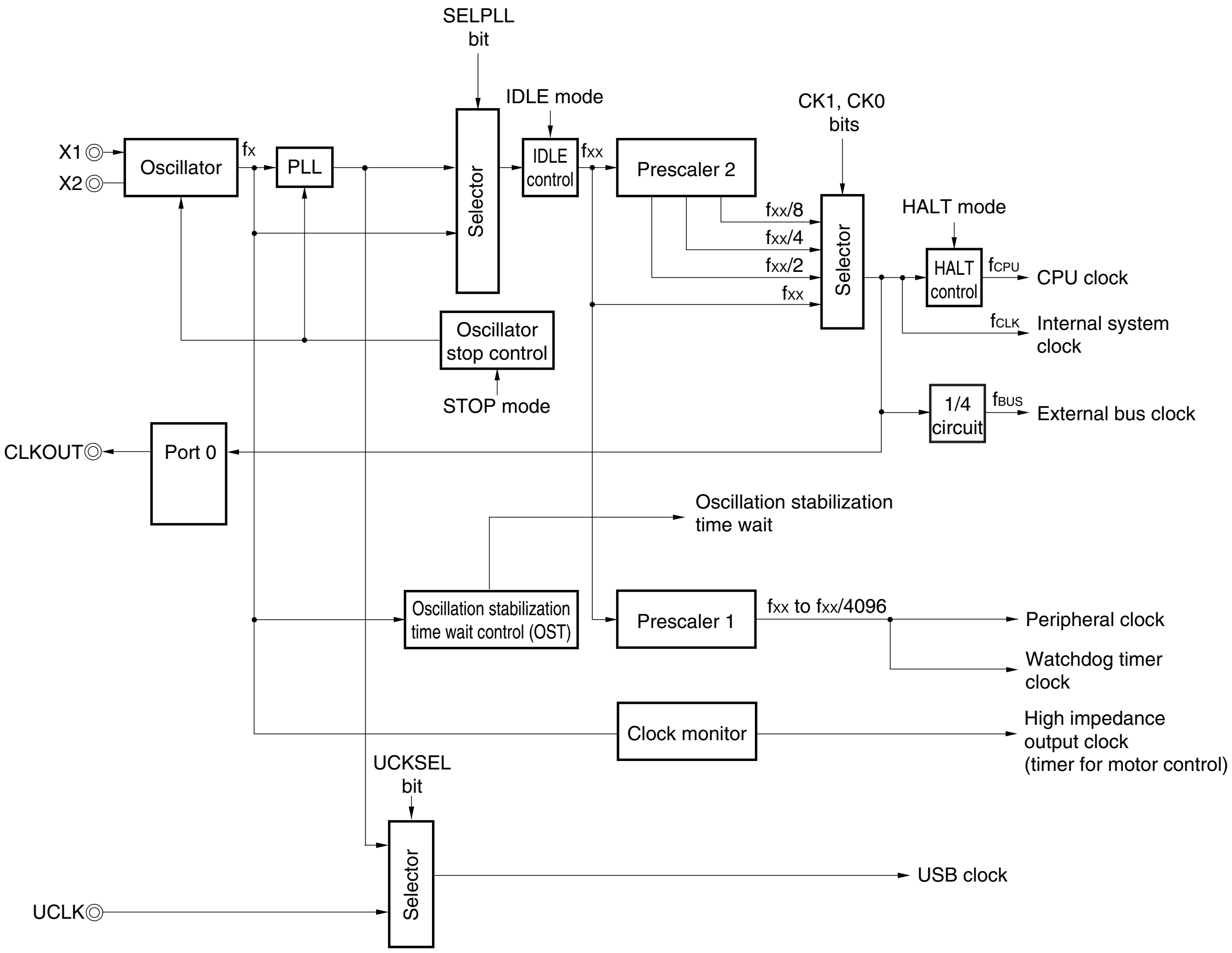
○ Oscillation stabilization time selection

Caution The oscillation guaranteed range is 10 to 12.5 MHz.

Remark f_X: Oscillation frequency
f_{XX}: System clock frequency

5.2 Configuration

Figure 5-1. Clock Generator



Caution Because f_{CPU} and f_{CLK} do not go through PLL immediately after reset, and $f_{XX}/8$ is selected by prescaler 2, if f_X = 10 MHz, f_{CPU} and f_{CLK} are 1.25 MHz.

Remark f_X: Oscillation frequency

f_{XX}: System clock frequency

f_{CPU}: CPU cock frequency

f_{CLK}: Internal system clock frequency

f_{BUS}: External bus clock frequency

Table 5-1. Operation Clock of Each Function Block

Function Block	Operation Clock
CPU	f_{CPU} (selected from f_{XX} to $f_{XX}/8$ by PCC register)
DMA, interrupt controller	f_{CLK} (selected from f_{XX} to $f_{XX}/8$ by PCC register)
TAA	$f_{XX}/2$
TAB	$f_{XX}/2$
TMT	$f_{XX}/2$
TMM	$f_{XX}/2$
Watchdog timer	$f_{XX}/1024$
UARTA	f_{UCLK} (selected from $f_{XX}/4$ to $f_{XX}/4096$ by UAnCTL1 register)
UARTB	$f_{XX}/2$
CSIF	f_{CCLK} (selected from $f_{XX}/16$ to $f_{XX}/512$ by CFnCTL1 register)
I^2C	$f_{XX}/8$
USB function	f_{USB} (Can be selected from the external clock input to the UCLK pin or the PLL output clock (96 MHz) divided by 2, by using the UCKSEL register.)
Bus control function	$f_{BUS} = f_{CLK}/4$
A/D converters 0, 1	f_{AD01} (selected from $f_{XX}/4$ to $f_{XX}/10$ by ADnOCKS register)
A/D converter 2	$f_{AD2} = f_{XX}/2$

Remarks 1. f_{CPU}: CPU cock frequency

f_{XX}: Peripheral clock frequency

f_{CLK}: Internal system clock frequency

f_{UCLK}: Base clock frequency of UARTA0 to UARTA2

f_{CCLK}: Base clock frequency of CSIF0 to CSIF2

f_{BUS}: External bus clock frequency

f_{AD01}: Base clock frequency of A/D converters 0 and 1

f_{AD2}: Operating clock frequency of A/D converter 2

2. n = 0, 1

(1) Oscillator

The main resonator oscillates the following frequencies (f_X):

- In PLL mode (×8 fixed): f_X = 10 to 12.5 MHz (f_{XX} = 80 to 100 MHz)
- In clock-through mode: f_X = 10 to 12.5 MHz (f_{XX} = 10 to 12.5 MHz)

(2) IDLE control

All functions other than the oscillator, PLL, clock monitor operation, CSIF in slave mode, low-voltage detector (LVI), and power-on-clear circuit (POC) are stopped.

(3) HALT control

Only the CPU clock (f_{CPU}) is stopped.

(4) PLL

This circuit multiplies the clock generated by the oscillator (f_X) by 8.

It operates in two modes: clock-through mode in which f_X is output as is by setting the SELPLL bit of the PLL control register (PLLCTL), and PLL mode in which a multiplied clock is output.

(5) Prescaler 1

This prescaler generates the clock (f_{XX} to f_{XX}/4096) to be supplied to on-chip peripheral functions.

(6) Prescaler 2

This circuit divides the system clock (f_{XX}).

The clock (f_{XX} to f_{XX}/8) to be supplied to the CPU clock (f_{CPU}) and internal system clock (f_{CLK}) is generated.

(7) Oscillation stabilization time wait control (OST)

This unit measures the time from when the clock generated by the oscillator was input until oscillation is stabilized. It also counts the PLL lockup time.

The count clock can be selected from $2^{15}/fx$ to $2^{18}/fx$.

(8) Clock monitor

The clock monitor samples the clock generated by the oscillator (f_X), by using the internal oscillation clock. When it detects stop of oscillation, output of the timer for motor control goes into a high-impedance state (for details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**).

5.3 Control Registers

The clock generator is controlled by the following six registers.

- PLL control register (PLLCTL)
- Processor clock control register (PCC)
- Power save control register (PSC)
- Power save mode register (PSMR)
- Oscillation stabilization time select register (OSTS)
- Clock monitor mode register (CLM)

(1) PLL control register (PLLCTL)

The PLLCTL register selects CPU operation clock.
This register can be read or written in 8-bit or 1-bit units.
Reset sets this register to 01H.

After reset: 01H R/W Address: FFFFF82CH

	7	6	5	4	3	2	<1>	0
PLLCTL	0	0	0	0	0	0	SELPLL	1

SELPLL	CPU operation clock selection
0	Clock-through mode
1	PLL mode

Cautions 1. Be sure to set bits 7 to 2 to "0" and set bit 0 to "1".

2. Setting the SELPLL bit to 1 is enabled only when the PLL clock frequency is stabilized. If the SELPLL bit is rewritten when the PLL clock frequency is not stabilized (during unlock), 0 is written to the bit. Therefore, be sure to confirm that the PLL mode has been set.
Use the following program for reference.

```
_loop:  set1   1, PLLCTL
        tst1   1, PLLCTL
        bz     _loop
(next instruction)
```

(2) Processor clock control register (PCC)

The PCC register is a special register. Data can be written to this register only in a combination of specific sequences (see **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 03H.

After reset: 03H R/W Address: FFFFF828H

	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	0	CK1	CK0

CK1	CK0	Clock selection (f_{CLK}/f_{CPU})
0	0	f_{XX}
0	1	$f_{XX}/2$
1	0	$f_{XX}/4$
1	1	$f_{XX}/8$

Cautions 1. Be sure to set bits 2 to 7 to “0”.

2. Set the PCC register to 00H after the PLL mode is selected (PLLCTL.SELPLL bit = 1).

(3) Power save control register (PSC)

The PSC register is an 8-bit register that controls the standby function and specifies the standby mode by setting the STB bit. The PSC register is a special register (see **3.4.8 Special registers**). Data can be written to this register only in a combination of specific sequences.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF1FEH

	7	6	5	<4>	3	2	<1>	0
PSC	0	0	0	INTM	0	0	STB	0

INTM	Standby mode control by maskable interrupt request (INTxx[Note 1])[Note 2]
0	Standby mode release by INTxx request enabled
1	Standby mode release by INTxx request disabled

STB	Sets operation mode
0	Normal mode
1	Standby mode

Notes 1. For details, see **Table 21-1 Interrupt Source List**.

2. Setting is valid only in the IDLE mode and STOP mode.

Cautions 1. Be sure to set bits 0, 2, 3, and 5 to 7 to “0”.

2. Before setting a standby mode by setting the STB bit to 1, be sure to set the PCC register to 03H and then set the STB bit to 1. Otherwise, the standby mode may not be set or released. After releasing the standby mode, change the value of the PCC register to the desired value.

3. To set the IDLE mode or STOP mode, set the PCC register to 03H, and the PSMR.PSM0 bit in that order and then set the STB bit to 1.

(4) Power save mode register (PSMR)

The PSMR register is an 8-bit register that controls the operation in the software standby mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	0	0	0	0	0	0	0	PSM0

PSM0	Specifies operation in software standby mode
0	IDLE mode
1	STOP mode

Cautions 1. Be sure to set bits 1 to 7 to “0”.

2. The PSM0 bit is valid only when the PSC.STB bit is 1.

(5) Oscillation stabilization time select register (OSTS)

The OSTS register selects the oscillation stabilization time until the oscillation stabilizes after the STOP mode is released by interrupt request.

This register can be read or written in 8-bit units.

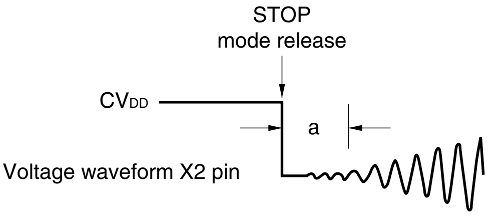
Reset sets this register to 05H.

After reset: 05H R/W Address: FFFFF6C0H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	OSTS3	OSTS2	OSTS1	OSTS0

OSTS3	OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time (f_X = 12.5 MHz)
0	1	0	1	$2^{15}/f_X$ (2.62 ms)
0	1	1	0	$2^{16}/f_X$ (5.24 ms)
0	1	1	1	$2^{17}/f_X$ (10.5 ms)
1	0	0	0	$2^{18}/f_X$ (21.0 ms)
Other than above				Setting prohibited

Cautions 1. The wait time does not include the time until the clock oscillation starts ("a" in the figure below) following release of the STOP mode.



2. The default value of the OSTS register after reset is 05H. If a 12.5 MHz resonator is used, therefore, the oscillation stabilization time is about 2.62 ms. Half the oscillation stabilization time is consumed by waiting for the lockup of PLL. Therefore, the actual stabilization time of the resonator is about 1.31 ms. When releasing reset, therefore, make sure that the oscillation stabilization time is secured during the active period of the reset signal. To release the STOP mode by an interrupt input other than a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by low-voltage detector (LVI), reset signal (POCRES) generation by power-on-clear circuit (POC)), the oscillation stabilization time is determined by the set value of the OSTS register. Therefore, set a time twice as long as that required for the resonator to stabilize to the OSTS register (because half the oscillation stabilization time is the stabilization time of PLL).

3. Be sure to set bits 4 to 7 to "0".

Remark f_X: Oscillation frequency

(6) Clock monitor mode register (CLM)

The CLM register sets clock monitor operation mode. The CLM register is a special register. It can be written only in a combination of specific sequences (see **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF870H

	7	6	5	4	3	2	1	0
CLM	0	0	0	0	0	0	0	CLME

CLME	Clock monitor operation control
0	Clock monitor operation disabled
1	Clock monitor operation enabled

Cautions 1. The CLME bit is cleared to 0 only after reset.

2. When the CLME bit = 1, the clock monitor function is forcibly stopped if the following condition is satisfied.

- **During oscillation stabilization time count after release of STOP mode**

3. When the CLME bit = 1, output of the timer for motor control goes into a high-impedance state if oscillation (f_X) stop is detected. See Figure 10-4 for the target timer output.

5.4 PLL Function

5.4.1 Overview

The CPU and the operating clock of the peripheral macro can be switched between output of the oscillation frequency multiplied by 8, and clock-through mode.

When PLL function is used: Input clock (f_X) = 10 to 12.5 MHz, output clock (f_{XX}) = 80 to 100 MHz
Clock-through mode: Input clock (f_X) = 10 to 12.5 MHz, output clock (f_{XX}) = 10 to 12.5 MHz

5.4.2 PLL mode

In the PLL mode, the oscillation frequency (f_X) is multiplied by 8 with the PLL to generate a system clock (f_{XX}).

In the PLL mode, the clock is input from the oscillator to the PLL. A clock at a stable frequency must be supplied to the internal circuit after the lapse of the lockup time (frequency stabilization time) during which the phase is locked at a specific frequency and oscillation is stabilized. In the V850E/IG4-H and V850E/IH4-H, the lockup time after release of reset is secured automatically.

Caution When a resonator of f_X = 12.5 MHz is used and if the oscillation stabilization time of that resonator must be 3 ms (MAX.), the reset input ($\overline{\text{RESET}}$ active) width must be 1.7 ms (MIN.).

5.4.3 Clock-through mode

In the clock-through mode, a system clock (f_{XX}) of the same frequency as the oscillation frequency (f_X) is generated.

5.5 Operation

5.5.1 Operation of each clock

The following table shows the operation status of each clock.

Table 5-2. Operation Status of Each Clock

Power Save Mode	Oscillator (f_X)	PLL	Internal System Clock (f_{CLK})	Peripheral Clock (f_{XX} to f_{XX}/4096)	External bus Clock (f_{BUS})	CPU Clock (f_{CPU})	USB clock (f_{USB})		Watchdog Timer Clock[Note 1]
							UCLK input	PLL output	
Normal operation	√	√	√	√	√	√	√	√	√
HALT mode	√	√	√	√	√	×	√	√	√
IDLE mode	√	√	×	×	×	×	√	√	×
In STOP mode and during oscillation stabilization time count after release of STOP mode	×[Note 2]	×[Note 2]	×	×	×	×	√	×	×
During $\overline{\text{RESET}}$ pin input[Note 3] and subsequent oscillation stabilization time count	√	× → √	√	×[Note 4]	×[Note 5]	√	√	× → √	×

Notes 1. The peripheral clock (f_{XX}/1024) is used as the watchdog timer clock.

2. Operation continues during on-chip debugging.

3. $\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by the watchdog timer, reset signal (LVIRES) generation by the low-voltage detector (LVI), or reset signal (POCRES) generation by the power-on-clear circuit (POC)

4. The output from the prescaler (PRS) in not performed.

5. The clock is not output from the CLKOUT pin.

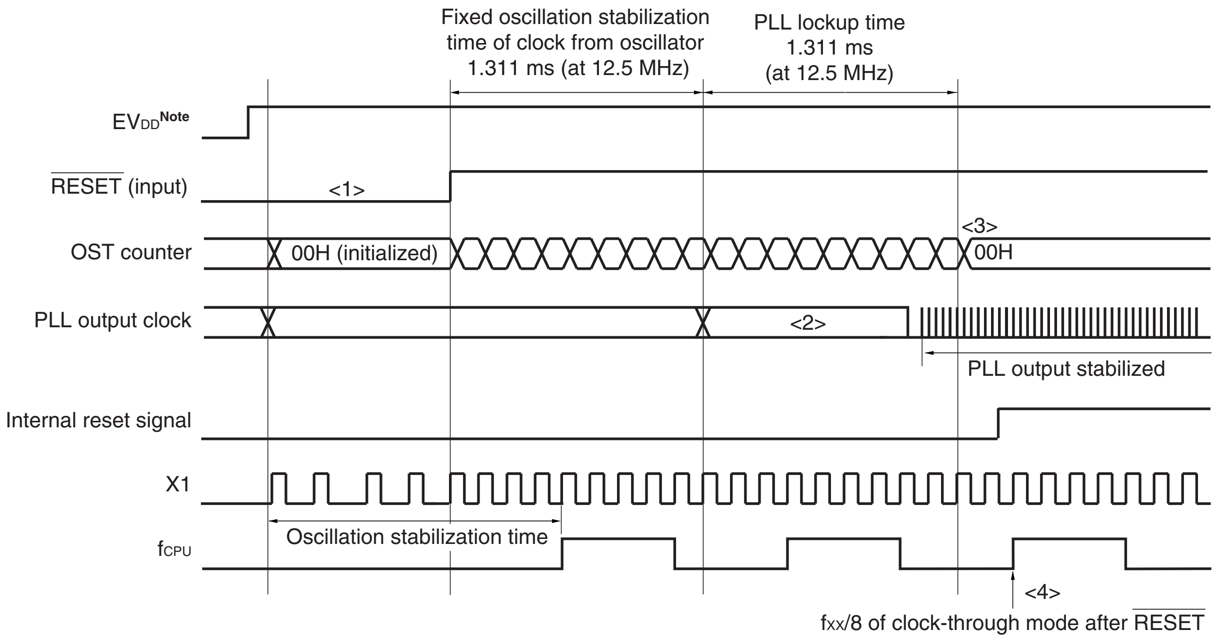
Remark √: Operating
×: Stopped

5.5.2 Clock output function

The clock output function is used to output the external bus clock (f_{BUS}) from the CLKOUT pin. The clock output function can be used when the internal system clock (f_{CLK}) is operable, as indicated by the check mark (√) in Table 5-2. The clock output function cannot be used when the internal system clock is stopped (as indicated by the cross (×) in Table 5-2).

5.5.3 Operation timing

(1) Power on (power-on reset)


Fixed oscillation stabilization time of clock from oscillator 1.311 ms (at 12.5 MHz)
PLL lockup time 1.311 ms (at 12.5 MHz)
EVDDNote
RESET (input)
<1>
OST counter
00H (initialized)
<3>
00H
PLL output clock
<2>
PLL output stabilized
Internal reset signal
X1
fCPU
Oscillation stabilization time
<4>
fxx/8 of clock-through mode after RESET

<1> The oscillator is activated during the RESET period that follows power application.
Make sure that the low-level width of the RESET signal is "Oscillation stabilization time of the used resonator − Fixed oscillation stabilization time" or more, taking the oscillation stabilization time into consideration.
PLL stops during the RESET period and fixed oscillation stabilization time.

<2> When the fixed oscillation stabilization time that elapses after the RESET signal is released expires, PLL stop is released, and counting the lockup time starts.

<3> PLL is locked when counting of the lockup time is over. The OST counter is initialized to 00H.

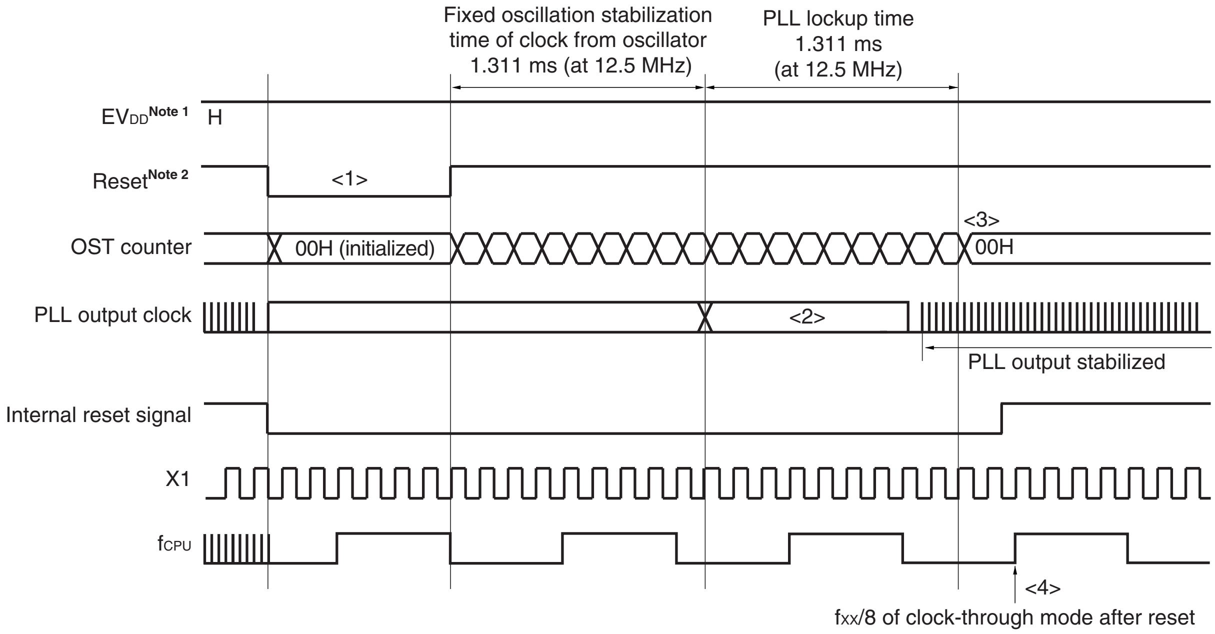
<4> When the lockup time expires, the CPU releases the reset signal and operates in the clock-through mode (fx). The CPU operation clock (f_{CPU}) is $f_{XX}/8$. The PLL mode can be set by software.

Note V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}
V850E/IH4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}

Cautions 1. The clock generated by the oscillator starts oscillating during the RESET period. After the RESET signal is released, a specific wait time (fixed oscillation stabilization time) elapses.

2. To avoid malfunction due to noise, do not change the division ratio of the CPU operation clock (f_{CPU}) by using the PCC register before setting the PLL mode. Before changing the division ratio, be sure to select the PLL mode.

(2) Reset input with power on


Fixed oscillation stabilization time of clock from oscillator 1.311 ms (at 12.5 MHz)
PLL lockup time 1.311 ms (at 12.5 MHz)
EVDD Note 1
H
Reset Note 2
<1>
OST counter
00H (initialized)
<3>
00H
PLL output clock
<2>
PLL output stabilized
Internal reset signal
X1
fCPU
<4>
fxx/8 of clock-through mode after reset

<1> The oscillator continues operating during the reset period.
PLL stops during the reset period and fixed oscillation stabilization time.

<2> When the fixed oscillation stabilization time that elapses after the reset signal is released expires, PLL stop is released, and counting the lockup time starts.

<3> PLL is locked when counting of the lockup time is over. The OST counter is initialized to 00H.

<4> When the lockup time expires, the CPU releases the reset signal and operates in the clock-through mode (f_X). The CPU operation clock (f_{CPU}) is f_{XX}/8. The PLL mode can be set by software.

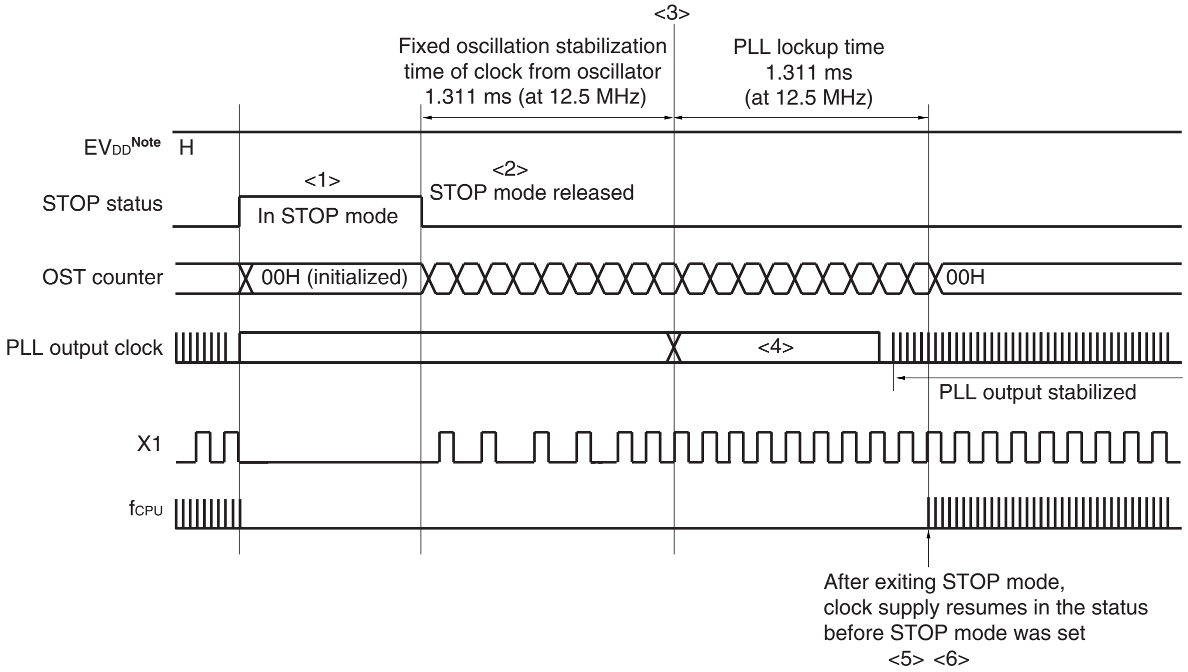
Notes 1. V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}
V850E/IH4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}

2. $\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by the watchdog timer, reset signal (LVIRES) generation by the low-voltage detector (LVI), or reset signal (POCRES) generation by the power-on-clear circuit (POC)

Cautions 1. The clock generated by the oscillator continues operating during a reset.
After the reset signal is released, a specific wait time (fixed oscillation stabilization time) elapses.

2. To avoid malfunction due to noise, do not change the division ratio of the CPU operation clock (f_{CPU}) by using the PCC register before setting the PLL mode. Before changing the division ratio, be sure to select the PLL mode.

(3) When releasing STOP mode by interrupt request


<3>
Fixed oscillation stabilization time of clock from oscillator 1.311 ms (at 12.5 MHz)
PLL lockup time 1.311 ms (at 12.5 MHz)
EVDDNote
H
STOP status
<1>
In STOP mode
<2>
STOP mode released
OST counter
00H (initialized)
00H
PLL output clock
<4>
PLL output stabilized
X1
fCPU
After exiting STOP mode, clock supply resumes in the status before STOP mode was set
<5> <6>

<1> When the STOP mode is set, both the oscillator and PLL stop.
At this time, PLL is stopped in the STOP mode. The OST counter is initialized.

<2> When the STOP mode is released, the oscillator is activated and the OST counter starts counting the oscillation stabilization time. At this time, PLL remains stopped.

<3> When a fixed oscillation stabilization time (1.311 ms) has elapsed, PLL starts operating. The clock generated by the oscillator must be stabilized before PLL starts operating. The actual oscillation stabilization time is “fixed oscillation stabilization time”. Take this into consideration when setting a value to the OSTS register.

<4> After a fixed oscillation stabilization time (1.311 ms) has elapsed, the lockup wait time starts. The remaining count time of the OST counter is the lockup wait time.

<5> When the lockup time of PLL is over, clock supply to the internal circuitry resumes in the status before the STOP mode was set.

<6> The operation to be performed when the STOP mode is released by a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by the low-voltage detector (LVI), reset signal (POCRES) generation by the power-on-clear circuit (POC)) is the same as that in **(1) Power on (power-on reset)** and **(2) Reset input with power on**.

Note V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}
V850E/IH4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}

5.6 Clock Monitor

(1) Clock monitor function

The clock monitor samples the clock generated by the oscillator, by using the internal oscillation clock. When it detects stop of oscillation, output of the timer for motor control goes into a high-impedance state (for details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**). The high-impedance state created by the clock monitor function is released by a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (POCRES) generation by the power-on-clear circuit (POC)) and the pin enters the status after reset.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER AA (TAA)

Timer AA (TAA) is a 16-bit timer/event counter.
The V850E/IG4-H and V850E/IH4-H incorporate TAA0 to TAA2.

6.1 Overview

The TAAn channels are outlined below (n = 0 to 2).

Table 6-1. TAAn Overview

Item	TAA0	TAA1	TAA2
Clock selection	8 ways	8 ways	8 ways
Capture trigger input pin	None	None	2
External event count input pin	None	None	1
External trigger input pin	None	None	1
Timer counter	1	1	1
Capture/compare register	2[Note]	2[Note]	2
Capture/compare match interrupt request signal	2[Note]	2[Note]	2
Overflow interrupt request signal	1	1	1
Timer output pin	2	2	2

Note Compare function only

6.2 Functions

The functions of TAAn that can be realized differ from one channel to another, as shown in the table below (n = 0 to 2).

Table 6-2. TAAn Functions

Function	TAA0	TAA1	TAA2
Interval timer	√	√	√
External event counter	×	×	√
External trigger pulse output	√Note 1	√Note 1	√
One-shot pulse output	√Note 1	√Note 1	√
PWM output	√	√	√
Free-running timer	√Note 2	√Note 2	√
Pulse width measurement	×	×	√
Timer tuning operation	√ (TAB0)	√ (TAB1)	×

Notes 1. This function can only be realized by using a software trigger; it cannot be realized by inputting an external trigger.

2. Compare function only

6.3 Configuration

TAAn includes the following hardware (n = 0 to 2).

Table 6-3. Configuration of TAAn

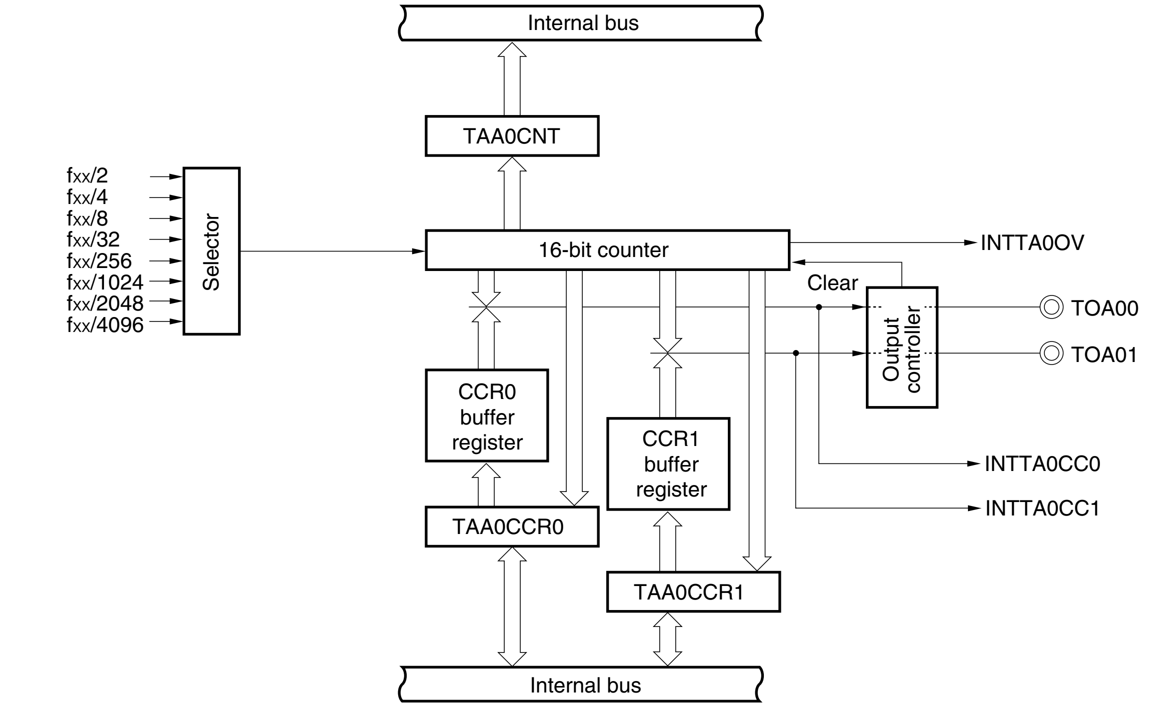
Item	Configuration
Timer register	16-bit counter × 1
Registers	TAAn capture/compare registers 0, 1 (TAAnCCR0, TAAnCCR1) TAAn counter read buffer register (TAAnCNT) CCR0 and CCR1 buffer registers
Timer input	2 in total (TIA20, TIA21 pins)Notes 1, 2
Timer output	6 in total (TOA00, TOA01, TOA10, TOA11, TOA20, TOA21 pins)Note 2
Control registers	TAAn control registers 0, 1 (TAAnCTL0, TAAnCTL1) TAAn I/O control registers 0 (TAAnIOC0) TAA2 I/O control registers 1, 2 (TAA2IOC1, TAA2IOC2) TAAn option registers 0 (TAAnOPT0)

Notes 1. Not provided for TAA0 and TAA1

2. The TIA20 pin functions alternately (alternate-function) as a capture trigger input, external event count input, external trigger input, and timer output (TOA20).
The TIA21 pin functions alternately as a capture trigger input and timer output (TOA21).

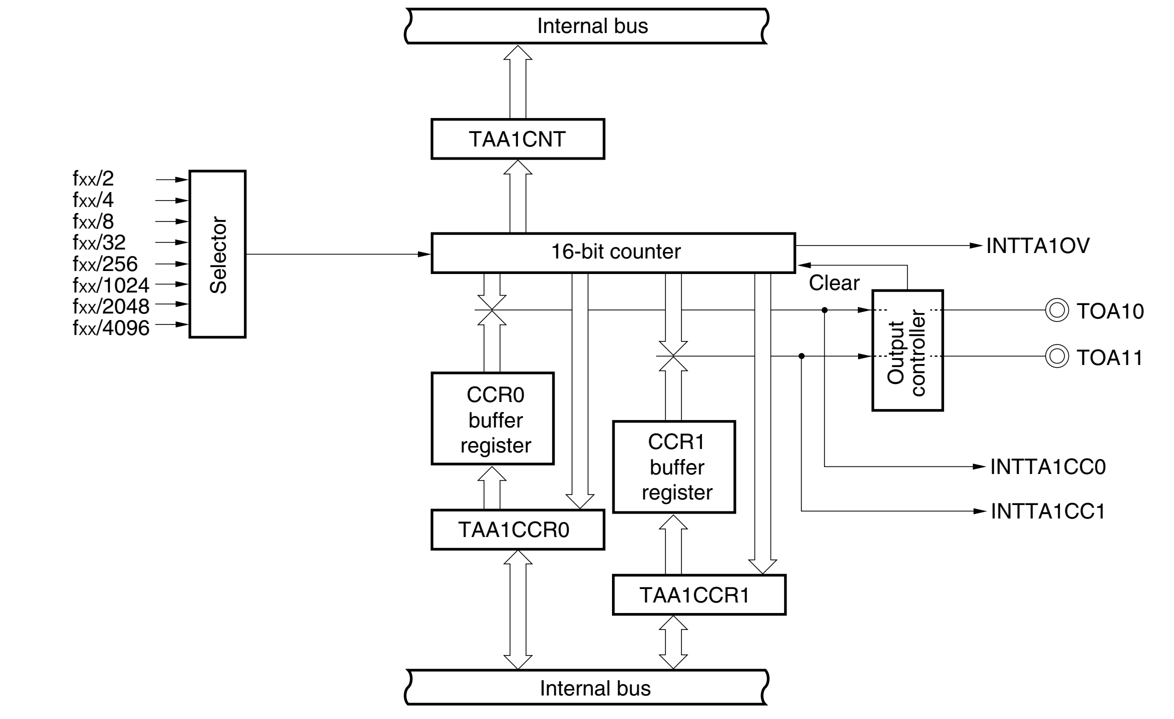
Remark n = 0 to 2

Figure 6-1. TAA0 Block Diagram



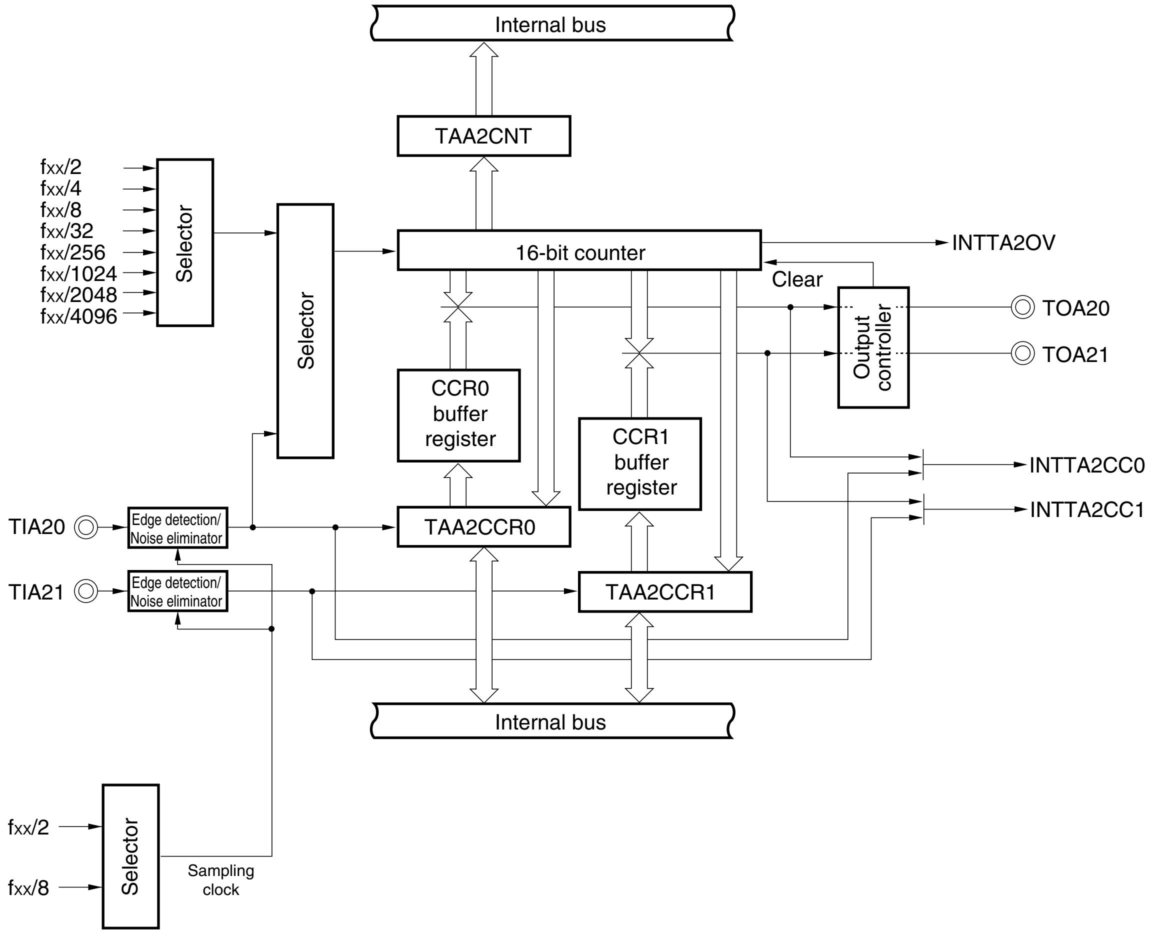
Remark fxx: Peripheral clock

Figure 6-2. TAA1 Block Diagram



Remark fxx: Peripheral clock

Figure 6-3. TAA2 Block Diagram



Remarks 1. fxx: Peripheral clock

2. For the noise eliminator, see **4.6 Noise Eliminator**.

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TAAnCNT register.

When the TAAnCTL0.TAAnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TAAnCNT register is read at this time, 0000H is read.

Reset sets the TAAnCE bit to 0.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TAAnCCR0 register is used as a compare register, the value written to the TAAnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTAnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TAAnCCR0 register is cleared to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TAAnCCR1 register is used as a compare register, the value written to the TAAnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, and the TAAnCCR1 register is cleared to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the TIA20 and TIA21 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TAAmIOC1 and TAAmIOC2 registers.

(5) Output controller

This circuit controls the output of the TOA00, TOA01, TOA10, TOA11, TOA20, and TOA21 pins. The output controller is controlled by the TAAnIOC0 registers.

(6) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

6.4 Registers

(1) TAAn control register 0 (TAAnCTL0)

The TAAnCTL0 register is an 8-bit register that controls the operation of TAAn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

The same value can always be written to the TAAnCTL0 register by software.

After reset: 00H R/W Address: TAA0CTL0 FFFFF660H, TAA1CTL0 FFFFF680H,
TAA2CTL0 FFFFF6A0H

	<7>	6	5	4	3	2	1	0
TAAnCTL0 (n = 0 to 2)	TAAnCE	0	0	0	0	TAAnCKS2	TAAnCKS1	TAAnCKS0

TAAnCE	TAAn operation control
0	TAAn operation disabled (TAAn reset asynchronously[Note])
1	TAAn operation enabled. TAAn operation start

TAAnCKS2	TAAnCKS1	TAAnCKS0	Internal count clock selection
0	0	0	fxx/2
0	0	1	fxx/4
0	1	0	fxx/8
0	1	1	fxx/32
1	0	0	fxx/256
1	0	1	fxx/1024
1	1	0	fxx/2048
1	1	1	fxx/4096

Note The TAAnOPT0.TAAnOVF bit and the 16-bit counter are reset simultaneously. Moreover, timer outputs (TOAn0 and TOAn1 pins) are reset to the TAAnIOC0 register set status at the same time as the 16-bit counter is reset.

Cautions 1. Set the TAAnCKS2 to TAAnCKS0 bits when the TAAnCE bit = 0.
When the value of the TAAnCE bit is changed from 0 to 1, the TAAnCKS2 to TAAnCKS0 bits can be set simultaneously.

2. Be sure to set bits 3 to 6 to “0”.

Remark fxx: Peripheral clock

(2) TAAn control register 1 (TAAnCTL1)

The TAAnCTL1 register is an 8-bit register that controls the TAAn operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAA0CTL1 FFFFF661H, TAA1CTL1 FFFFF681H,
TAA2CTL1 FFFFF6A1H

	7	6	5	4	3	2	1	0
TAAnCTL1	TAAaSYE[Note 1]	TAAnEST	TAA2EEE[Note 2]	0	0	TAAnMD2	TAAnMD1	TAAnMD0

(n = 0 to 2
a = 0, 1)

TAAaSYE[Note 1]	Operation mode selection
0	TAAa single mode
1	Tuning operation mode (see **10.4.5**)
TAAa can be used only as an A/D conversion start trigger factor of A/D converters 0 and 1 during the tuning operation. In the tuning operation mode, this bit always operates in synchronization with TABa.	

TAAnEST	Software trigger control
0	–
1	Generates a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TAAnEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TAAnEST bit as the trigger.
The read value of the TAAnEST bit is always 0.	

Notes 1. This bit can be set only in TAA0 and TAA1. Be sure to set bit 7 of TAA2 to “0”.
For details of tuning operation mode, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

2. This bit can be set only in TAA2. Be sure to set bits 5 of TAA0 and TAA1 to “0”.

(2/2)

TAA2EEE[Note 1]	Count clock selection
0	Disable operation with external event count input (TIA20 pin). (Perform counting with the count clock selected by the TAA2CTL0.TAA2CKS0 to TAA2CTL0.TAA2CKS2 bits.)
1	Enable operation[Note 2] with external event count input (TIA20 pin). (Perform counting at every valid edge of the external event count input signal (TIA20 pin).)
The TAA2EEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TAAnMD2	TAAnMD1	TAAnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode[Note 3]
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode[Note 3]
1	1	1	Setting prohibited

Notes 1. This bit can be set only in TAA2. Be sure to set bits 5 of TAA0 and TAA1 to “0”.

2. Set the valid edge selection of capture trigger input (TIA20 pin) and external trigger input (TIA20 pin) to “No edge detection”.

3. The external event count mode and pulse width measurement mode cannot be specified for TAA0 and TAA1.

Cautions 1. The TAAnEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. External event count input is selected in the external event count mode regardless of the value of the TAA2EEE bit.

3. Set the TAAaSYE, TAA2EEE, and TAAnMD2 to TAAnMD0 bits when the TAAnCTL0.TAAnCE bit = 0. (The same value can be written when the TAAnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TAAnCE bit = 1. If rewriting was mistakenly performed, clear the TAAnCE bit to 0 and then set the bits again.

4. Be sure to set bits 3 and 4 to “0”.

(3) TAAn I/O control register 0 (TAAnIOC0)

The TAAnIOC0 register is an 8-bit register that controls the timer output (TOAn0, TOAn1 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAA0IOC0 FFFFF662H, TAA1IOC0 FFFFF682H
TAA2IOC0 FFFFF6A2H

	7	6	5	4	3	<2>	1	<0>
TAAnIOC0	0	0	0	0	TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0

(n = 0 to 2, a = 0, 1)

TAAnOL1	TOAn1 pin output level setting[Note]
0	TOAn1 pin starts output at high level.
1	TOAn1 pin starts output at low level.

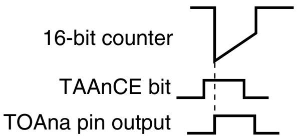
TAAnOE1	TOAn1 pin output setting
0	Timer output prohibited • Low level is output from the TOAn1 pin when the TAAnOL1 bit = 0. • High level is output from the TOAn1 pin when the TAAnOL1 bit = 1.
1	Timer output enabled (A pulse is output from the TOAn1 pin.)

TAAnOL0	TOAn0 pin output level setting[Note]
0	TOAn0 pin starts output at high level.
1	TOAn0 pin starts output at low level.

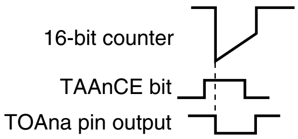
TAAnOE0	TOAn0 pin output setting
0	Timer output prohibited • Low level is output from the TOAn0 pin when the TAAnOL0 bit = 0. • High level is output from the TOAn0 pin when the TAAnOL0 bit = 1.
1	Timer output enabled (A pulse is output from the TOAn0 pin.)

Note The output level of the timer output pins (TOAn0 and TOAn1) specified by the TAAnOLa bit is shown below.

• When TAAnOLa bit = 0



• When TAAnOLa bit = 1



(2/2)

Cautions 1. If the setting of the TAAnIOC0 register is changed when TOAn0 and TOAn1 are set in the output mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.

2. Rewrite the TAAnOL1, TAAnOE1, TAAnOL0, and TAAnOE0 bits when the TAAnCTL0.TAAnCE bit = 0. (The same value can be written when the TAAnCE bit = 1.) If rewriting was mistakenly performed, clear the TAAnCE bit to 0 and then set the bits again.

3. Even if the TAAnOL0 or TAAnOL1 bit is manipulated when the TAAnCE, TAAnOE0, and TAAnOE1 bits are 0, the output level of the TOAn0 and TOAn1 pins changes.

(4) TAA2 I/O control register 1 (TAA2IOC1)

The TAA2IOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TIA20, TIA21 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF6A3H

	7	6	5	4	3	2	1	0
TAA2IOC1	0	0	0	0	TAA2IS3	TAA2IS2	TAA2IS1	TAA2IS0

TAA2IS3	TAA2IS2	Capture trigger input signal (TIA21 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TAA2IS1	TAA2IS0	Capture trigger input signal (TIA20 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TAA2IS3 to TAA2IS0 bits when the TAA2CTL0.TAA2CE bit = 0. (The same value can be written when the TAA2CE bit = 1.) If rewriting was mistakenly performed, clear the TAA2CE bit to 0 and then set the bits again.

2. The TAA2IS3 to TAA2IS0 bits are valid only in the free-running timer mode (only when the TAA2OPT0.TAA2CCS1 and TAA2OPT0.TAA2CCS0 bits = 11) and the pulse width measurement mode. In all other modes, a capture operation is not possible.

(5) TAA2 I/O control register 2 (TAA2IOC2)

The TAA2IOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (TIA20 pin) and external trigger input signal (TIA20 pin).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF6A4H

	7	6	5	4	3	2	1	0
TAA2IOC2	0	0	0	0	TAA2EES1	TAA2EES0	TAA2ETS1	TAA2ETS0

TAA2EES1	TAA2EES0	External event count input signal (TIA20 pin) valid edge setting
0	0	No edge detection (external event count invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TAA2ETS1	TAA2ETS0	External trigger input signal (TIA20 pin) valid edge setting
0	0	No edge detection (external trigger invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TAA2EES1, TAA2EES0, TAA2ETS1, and TAA2ETS0 bits when the TAA2CTL0.TAA2CE bit = 0. (The same value can be written when the TAA2CE bit = 1.) If rewriting was mistakenly performed, clear the TAA2CE bit to 0 and then set the bits again.

2. The TAA2EES1 and TAA2EES0 bits are valid only when the TAA2CTL1.TAA2EEE bit = 1 or when the external event count mode (the TAA2CTL1.TAA2MD2 to TAA2CTL1.TAA2MD0 bits = 001) has been set.

3. The TAA2ETS1 and TAA2ETS0 bits are valid only in the external trigger pulse output mode or one-shot pulse output mode.

(6) TAAn option register 0 (TAAnOPT0)

The TAAnOPT0 register is an 8-bit register that sets the capture/compare operation and detects overflow.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TAA0OPT0 FFFFF665H, TAA1OPT0 FFFFF685H,
TAA2OPT0 FFFFF6A5H

	7	6	5	4	3	2	1	<0>
TAAnOPT0 (n = 0 to 2)	0	0	TAA2CCS1[Note]	TAA2CCS0[Note]	0	0	0	TAAnOVF

TAA2CCS1[Note]	TAA2CCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TAA2CTL0.TAA2CE bit = 0)
The TAA2CCS1 bit setting is valid only in the free-running timer mode.	

TAA2CCS0[Note]	TAA2CCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TAA2CTL0.TAA2CE bit = 0)
The TAA2CCS0 bit setting is valid only in the free-running timer mode.	

TAAnOVF	TAAn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	0 is written to TAAnOVF bit or TAAnCTL0.TAAnCE bit = 0
• The TAAnOVF bit is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode. • An overflow interrupt request signal (INTTAnOV) is generated at the same time that the TAAnOVF bit is set to 1. The INTTAnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. • The TAAnOVF bit is not cleared to 0 even when the TAAnOVF bit or the TAAnOPT0 register are read when the TAAnOVF bit = 1. • Before clearing the TAAnOVF bit to 0 after generation of the INTTAnOV signal, be sure to confirm (by reading) that the TAAnOVF bit is set to 1. • The TAAnOVF bit can be both read and written, but the TAAnOVF bit cannot be set to 1 by software. Writing 1 has no effect on the operation of TAAn.	

Note This bit can be set only in TAA2. Be sure to set bits 4 and 5 of TAA0 and TAA1 to “0”.

Cautions 1. Rewrite the TAA2CCS1 and TAA2CCS0 bits when the TAA2CE bit = 0. (The same value can be written when the TAA2CE bit = 1.) If rewriting was mistakenly performed, clear the TAA2CE bit to 0 and then set the bits again.

2. Be sure to set bits 1 to 3, 6, and 7 to “0”.

(7) TAAn capture/compare register 0 (TAAnCCR0)

The TAA2CCR0 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TAAkCCR0 register is a 16-bit register that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TAA2OPT0.TAA2CCS0 bit. In the pulse width measurement mode, the TAA2CCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TAAnCCR0 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Remark n = 0 to 2, k = 0, 1

After reset: 0000H R/W Address: TAA0CCR0 FFFFF666H, TAA1CCR0 FFFFF686H,
TAA2CCR0 FFFFF6A6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAAnCCR0 (n = 0 to 2)																

(a) Function as compare register

The TAAnCCR0 register can be rewritten even when the TAAnCTL0.TAAnCE bit = 1.

The set value of the TAAnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTAnCC0) is generated. If TOAn0 pin output is enabled at this time, the output of the TOAn0 pin is inverted.

When the TAAnCCR0 register is used as a cycle register in the interval timer mode, external trigger pulse output mode, one-shot pulse output mode, and PWM output mode or the TAA2CCR0 register is used as a cycle register in external event count mode, the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

The compare register is not cleared by setting the TAAnCTL0.TAAnCE bit to 0.

(b) Function as capture register

When the TAA2CCR0 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TAA2CCR0 register if the valid edge of the capture trigger input pin (TIA20 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TAA2CCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIA20 pin) is detected.

Even if the capture operation and reading the TAA2CCR0 register conflict, the correct value of the TAA2CCR0 register can be read.

The capture register is cleared by setting the TAA2CTL0.TAA2CE bit to 0.

Remark n = 0 to 2

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter[Note 1]	Compare register	Anytime write
External trigger pulse output[Note 2]	Compare register	Batch write[Note 3]
One-shot pulse output[Note 2]	Compare register	Anytime write
PWM output	Compare register	Batch write[Note 3]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TAA2 only

2. When using TAA0 and TAA1, this function can only be realized by using a software trigger; it cannot be realized by inputting an external trigger.

3. Writing to the TAAnCCR1 register is the trigger.

Remark For anytime write and batch write, see **6.6 (2) Anytime write and batch write**.

(8) TAAn capture/compare register 1 (TAAnCCR1)

The TAA2CCR1 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TAAkCCR1 register is a 16-bit register that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TAA2OPT0.TAA2CCS1 bit. In the pulse width measurement mode, the TAA2CCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TAAnCCR1 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Remark n = 0 to 2, k = 0, 1

After reset: 0000H R/W Address: TAA0CCR1 FFFFF668H, TAA1CCR1 FFFFF688H,
TAA2CCR1 FFFFF6A8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAAnCCR1 (n = 0 to 2)																

(a) Function as compare register

The TAAnCCR1 register can be rewritten even when the TAAnCTL0.TAAnCE bit = 1.

The set value of the TAAnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAnCC1) is generated. If TOAn1 pin output is enabled at this time, the output of the TOAn1 pin is inverted.

The compare register is not cleared by setting the TAAnCTL0.TAAnCE bit to 0.

(b) Function as capture register

When the TAA2CCR1 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TAA2CCR1 register if the valid edge of the capture trigger input pin (TIA21 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TAA2CCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIA21 pin) is detected.

Even if the capture operation and reading the TAA2CCR1 register conflict, the correct value of the TAA2CCR1 register can be read.

The capture register is cleared by setting the TAA2CTL0.TAA2CE bit to 0.

Remark n = 0 to 2

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter[Note 1]	Compare register	Anytime write
External trigger pulse output[Note 2]	Compare register	Batch write[Note 3]
One-shot pulse output[Note 2]	Compare register	Anytime write
PWM output	Compare register	Batch write[Note 3]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TAA2 only

2. When using TAA0 and TAA1, this function can only be realized by using a software trigger. It cannot be realized by inputting an external trigger.

2. Writing to the TAAnCCR1 register is the trigger.

Remark For anytime write and batch write, see **6.6 (2) Anytime write and batch write**.

(9) TAAn counter read buffer register (TAAnCNT)

The TAAnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TAAnCTL0.TAAnCE bit = 1, the count value of the 16-bit timer can be read.

This register is read-only, in 16-bit units.

The value of the TAAnCNT register is cleared to 0000H when the TAAnCE bit = 0. If the TAAnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The value of the TAAnCNT register is cleared to 0000H after reset, and the TAAnCE bit is cleared to 0.

After reset: 0000H R Address: TAA0CNT FFFFF66AH, TAA1CNT FFFFF68AH,
TAA2CNT FFFFF6AAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAAnCNT (n = 0 to 2)																

6.5 Timer Output Operations

The following table shows the operations and output levels of the TOAn0 and TOAn1 pins.

Table 6-6. Timer Output Control in Each Mode

Operation Mode	TOAn1 Pin	TOAn0 Pin
Interval timer mode	PWM output	
External event count mode	None	
External trigger pulse output mode	External trigger pulse output	PWM output
One-shot pulse output mode	One-shot pulse output	
PWM output mode	PWM output	
Free-running timer mode	PWM output (only when compare function is used)	
Pulse width measurement mode	None	

Remark n = 0 to 2

Table 6-7. Truth Table of TOAn0 and TOAn1 Pins Under Control of Timer Output Control Bits

TAAnIOC0.TAAnOLa Bit	TAAnIOC0.TAAnOEa Bit	TAAnCTL0.TAAnCE Bit	Level of TOAna Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark n = 0 to 2, a = 0, 1

6.6 Operation

The functions of TAAn that can be achieved differ from one channel to another. The functions of each channel are shown below.

Table 6-8. TAA0 and TAA1 Specifications in Each Mode

Operation	Software Trigger Bit	External Trigger Input	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	None			
External trigger pulse output mode[Note]	Valid	Invalid	Compare only	Batch write
One-shot pulse output mode[Note]	Valid	Invalid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Compare only	Anytime write
Pulse width measurement mode	None			

Remarks 1. TAAa does not have timer input pins (TIAa0, TIAa1). It has interrupt request signals (INTTAaCC0, INTTAaCC1) on a match between the value of the 16-bit counter and the values of the TAAaCCR0 and TAAaCCR1 registers.

2. TAAa has a function to execute tuning with TABa. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

3. a = 0, 1

Note When using the external trigger pulse output mode and one-shot pulse output mode, select the internal clock as the count clock (by clearing the TAAaCTL1.TAAaEEE bit to 0).

Table 6-9. TAA2 Specifications in Each Mode

Operation	TAA2CTL1.TAA2EST Bit (Software Trigger Bit)	TIA20 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode[Note 1]	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode[Note 2]	Valid	Valid	Compare only	Batch write
One-shot pulse output mode[Note 2]	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switchable	Anytime write
Pulse width measurement mode[Note 2]	Invalid	Invalid	Capture only	Not applicable

Notes 1. When using the external event count mode, set the TIA20 pin capture trigger input valid edge selection to “No edge detection”. (Clear the TAA2IOC1.TAA2IS1 and TAA2IOC1.TAA2IS0 bits to 00.)

2. When using the external trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode, select the internal clock as the count clock (by clearing the TAA2CTL1.TAA2EEE bit to 0).

(1) Counter basic operation

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

Remark n = 0 to 2

(a) Counter start operation

- In external event count mode
  When the TAA2CTL0.TAA2CE bit is set from 0 to 1, the 16-bit counter is set to 0000H.
  After that, it counts up to 0001H, 0002H, 0003H, … each time the valid edge of external event count input (TIA20) is detected.
- In modes other than the above
  Starts counting from the default value FFFFH.
  It counts up from FFFFH to 0000H, 0001H, 0002H, 0003H, and so on.

(b) Clear operation

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and is cleared, and when its value is captured and cleared. The counting operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTAnCC0 and INTTAnCC1 interrupt signals are not generated.

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running timer mode or pulse width measurement mode. If the counter overflows, the TAAnOPT0.TAAnOVF bit is set to 1 and an interrupt request signal (INTTAnOV) is generated. Note that the INTTAnOV signal is not generated under the following conditions.

- Immediately after a counting operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured and cleared in the pulse width measurement mode and the counter counts up from FFFFH to 0000H

Caution **After the overflow interrupt request signal (INTTAnOV) has been generated, be sure to check that the overflow flag (TAAnOVF bit) is set to 1.**

(d) Counter read operation during counting operation

The value of the 16-bit counter of TAAn can be read by using the TAAnCNT register during the count operation. When the TAAnCTL0.TAAnCE bit = 1, the value of the 16-bit counter can be read by reading the TAAnCNT register. When the TAAnCTL0.TAAnCE bit = 0, the 16-bit counter is FFFFH and the TAAnCNT register is 0000H.

(e) Interrupt operation

TAAn generates the following three types of interrupt request signals.

- INTTAnCC0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TAAnCCR0 register.
- INTTAnCC1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TAAnCCR1 register.
- INTTAnOV interrupt: This signal functions as an overflow interrupt request signal.

(2) Anytime write and batch write

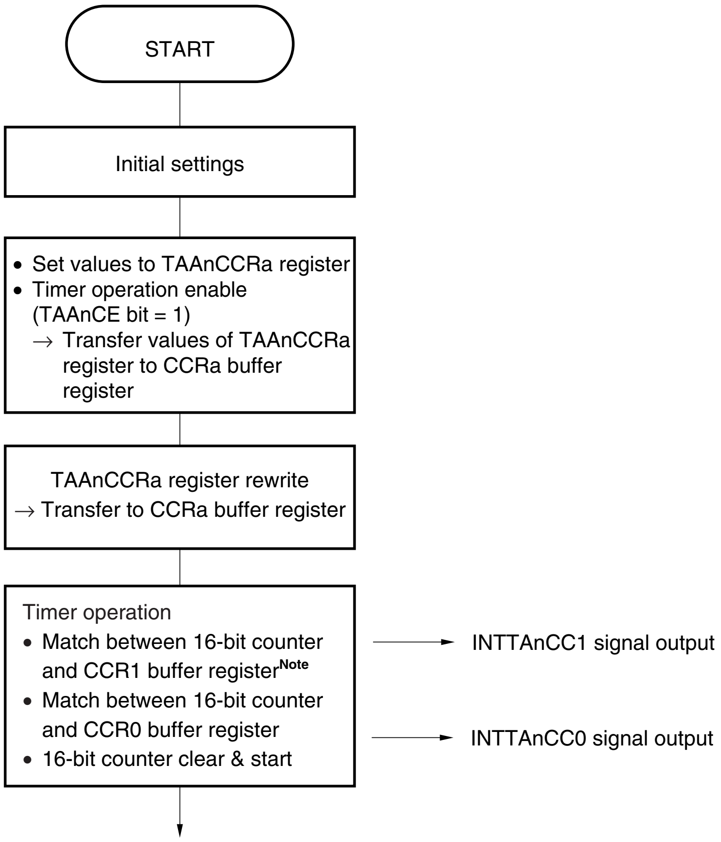
The TAAnCCR0 and TAAnCCR1 registers in TAAn can be rewritten during timer operation (TAAnCTL0.TAAnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 and CCR1 buffer registers differs depending on the mode.

(a) Anytime write

In this mode, data is transferred at any time from the TAAnCCR0 and TAAnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation.

Remark n = 0 to 2

Figure 6-4. Flowchart of Basic Operation for Anytime Write

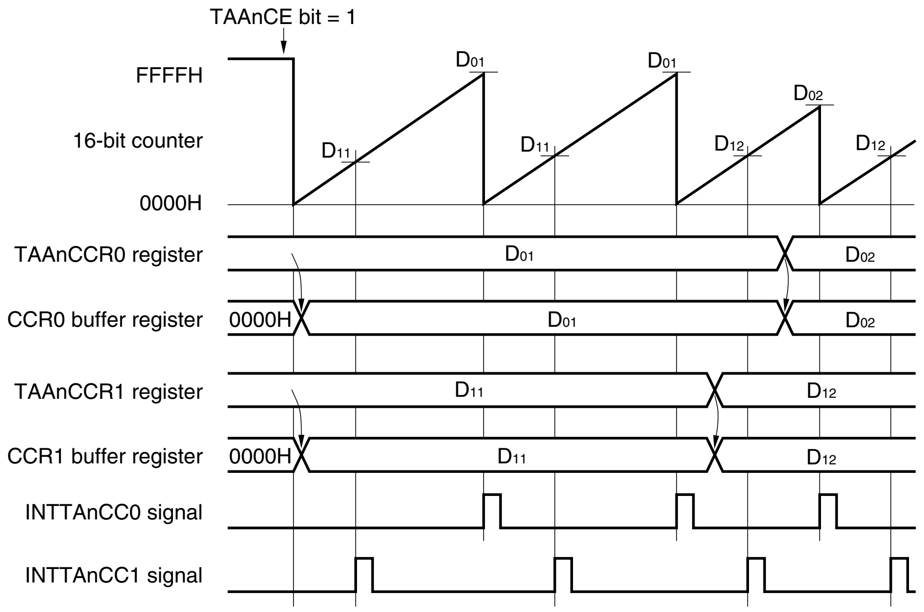

START
Initial settings
• Set values to TAAnCCRa register
• Timer operation enable (TAAnCE bit = 1)
→ Transfer values of TAAnCCRa register to CCRa buffer register
TAAnCCRa register rewrite
→ Transfer to CCRa buffer register
Timer operation
• Match between 16-bit counter and CCR1 buffer register[Note]
• Match between 16-bit counter and CCR0 buffer register
• 16-bit counter clear & start
INTTAnCC1 signal output
INTTAnCC0 signal output

Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Remarks 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0 to 2
a = 0, 1

Figure 6-5. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Set values of the TAAnCCR0 register

D_{11}, D_{12}: Set values of the TAAnCCR1 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0 to 2

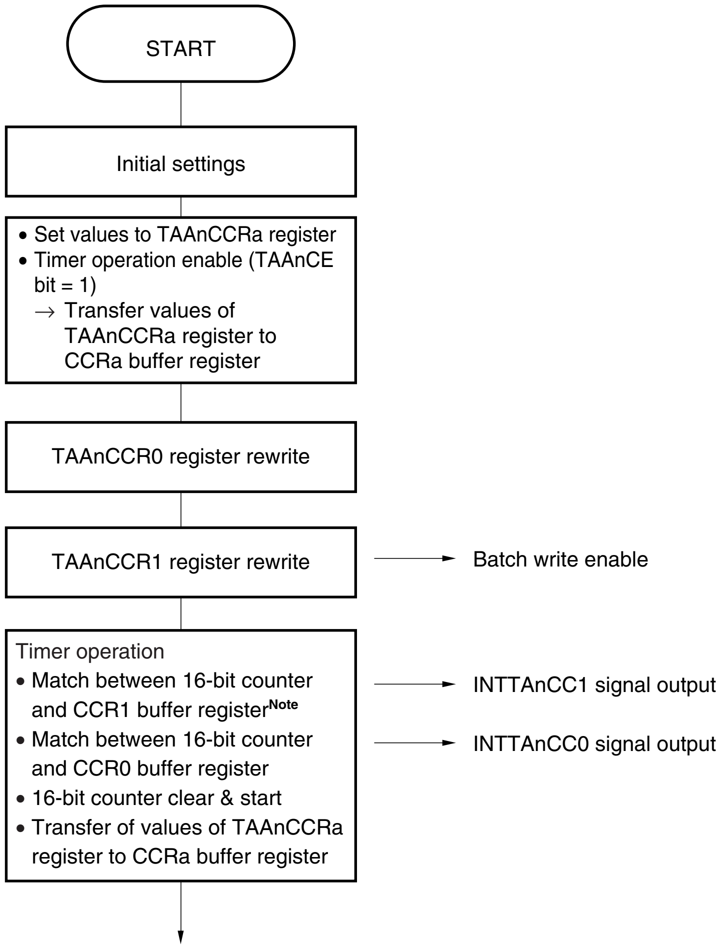
(b) Batch write

In this mode, data is transferred all at once from the TAAnCCR0 and TAAnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TAAnCCR1 register. Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TAAnCCR1 register.

In order for the set value when the TAAnCCR0 and TAAnCCR1 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 and CCR1 buffer registers), it is necessary to rewrite the TAAnCCR0 register and then write to the TAAnCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. Therefore, the values of the TAAnCCR0 and TAAnCCR1 registers are transferred to the CCR0 and CCR1 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus even when wishing only to rewrite the value of the TAAnCCR0 register, also write the same value (same as preset value of the TAAnCCR1 register) to the TAAnCCR1 register.

Remark n = 0 to 2

Figure 6-6. Flowchart of Basic Operation for Batch Write


START
Initial settings
• Set values to TAAnCCRa register
• Timer operation enable (TAAnCE bit = 1)
→ Transfer values of TAAnCCRa register to CCRa buffer register
TAAnCCR0 register rewrite
TAAnCCR1 register rewrite
Batch write enable
Timer operation
• Match between 16-bit counter and CCR1 buffer registerNote
• Match between 16-bit counter and CCR0 buffer register
• 16-bit counter clear & start
• Transfer of values of TAAnCCRa register to CCRa buffer register
INTTAnCC1 signal output
INTTAnCC0 signal output

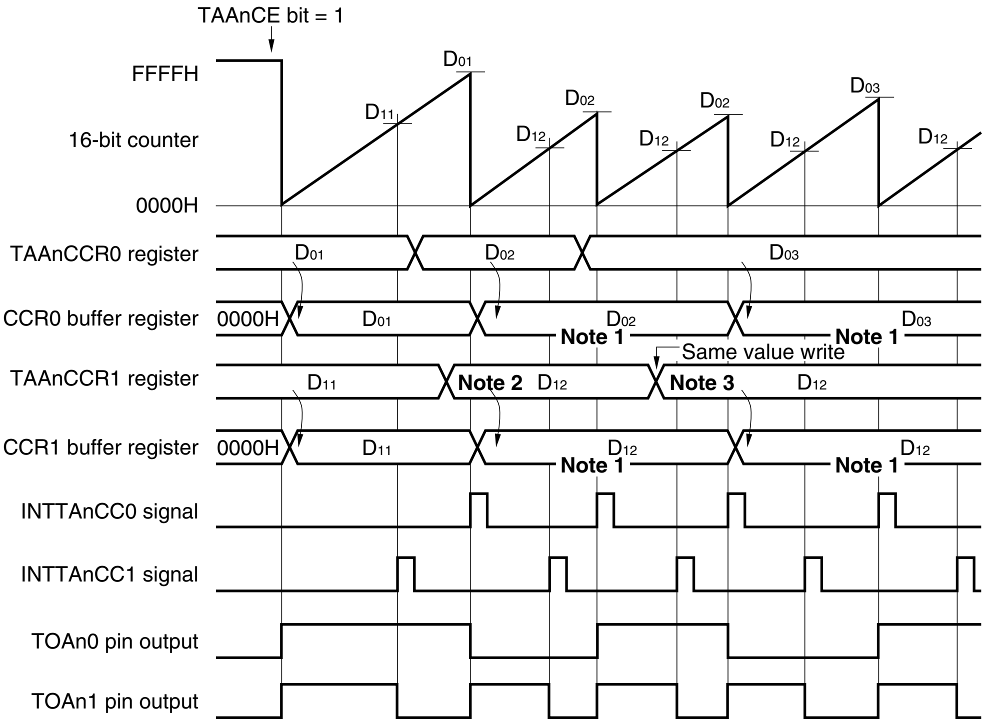
Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Caution Writing to the TAAnCCR1 register includes enabling of batch write. Thus, rewrite the TAAnCCR1 register after rewriting the TAAnCCR0 register.

Remarks 1. The above flowchart illustrates an example of the operation in the PWM output mode.

2. n = 0 to 2
a = 0, 1

Figure 6-7. Timing of Batch Write



Notes 1. Because the TAAnCCR1 register was not rewritten, D_{03} is not transferred.

2. Because the TAAnCCR1 register has been written (D_{12}), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TAAnCCR0 register (D_{01}).

3. Because the TAAnCCR1 register has been written (D_{12}), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TAAnCCR0 register (D_{02}).

Remarks 1. D_{01}, D_{02}, D_{03}: Set values of TAAnCCR0 register
D_{11}, D_{12}: Set values of TAAnCCR1 register

2. The above timing chart illustrates the operation in the PWM output mode as an example.

3. n= 0 to 2

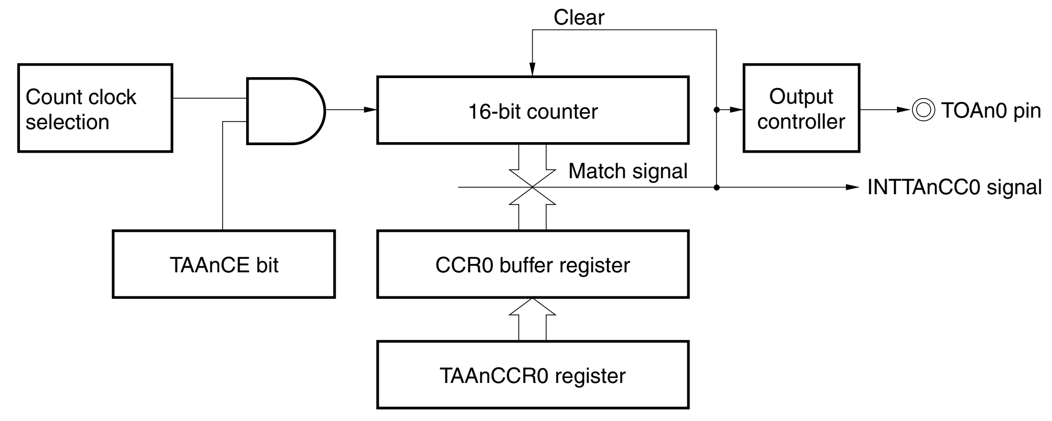
6.6.1 Interval timer mode (TAAnMD2 to TAAnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTTAnCC0) is generated at the interval set by the TAAnCCR0 register if the TAAnCTL0.TAAnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOAn0 pin.

The TAAnCCR1 register is not used in the interval timer mode. However, the set value of the TAAnCCR1 register is transferred to the CCR1 buffer register, and when the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAnCC1) is generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTAnCC1 signal is generated, can be output from the TOAn1 pin.

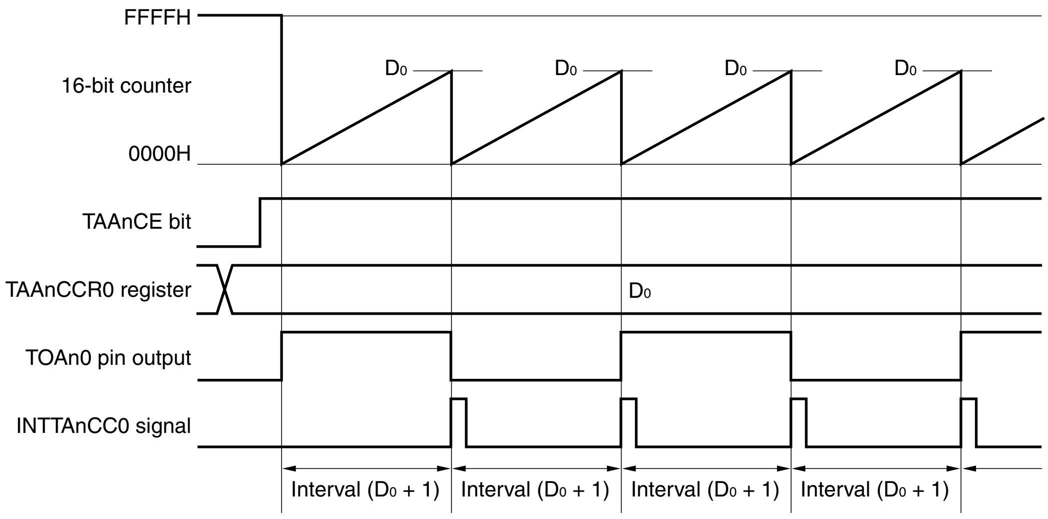
The value of the TAAnCCR0 and TAAnCCR1 registers can be rewritten even while the timer is operating.

Figure 6-8. Configuration of Interval Timer



Remark n = 0 to 2

Figure 6-9. Basic Timing of Operation in Interval Timer Mode



Remark n = 0 to 2

When the TAAnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TOAn0 pin is inverted. Additionally, the set value of the TAAnCCR0 register is transferred to the CCR0 buffer register.

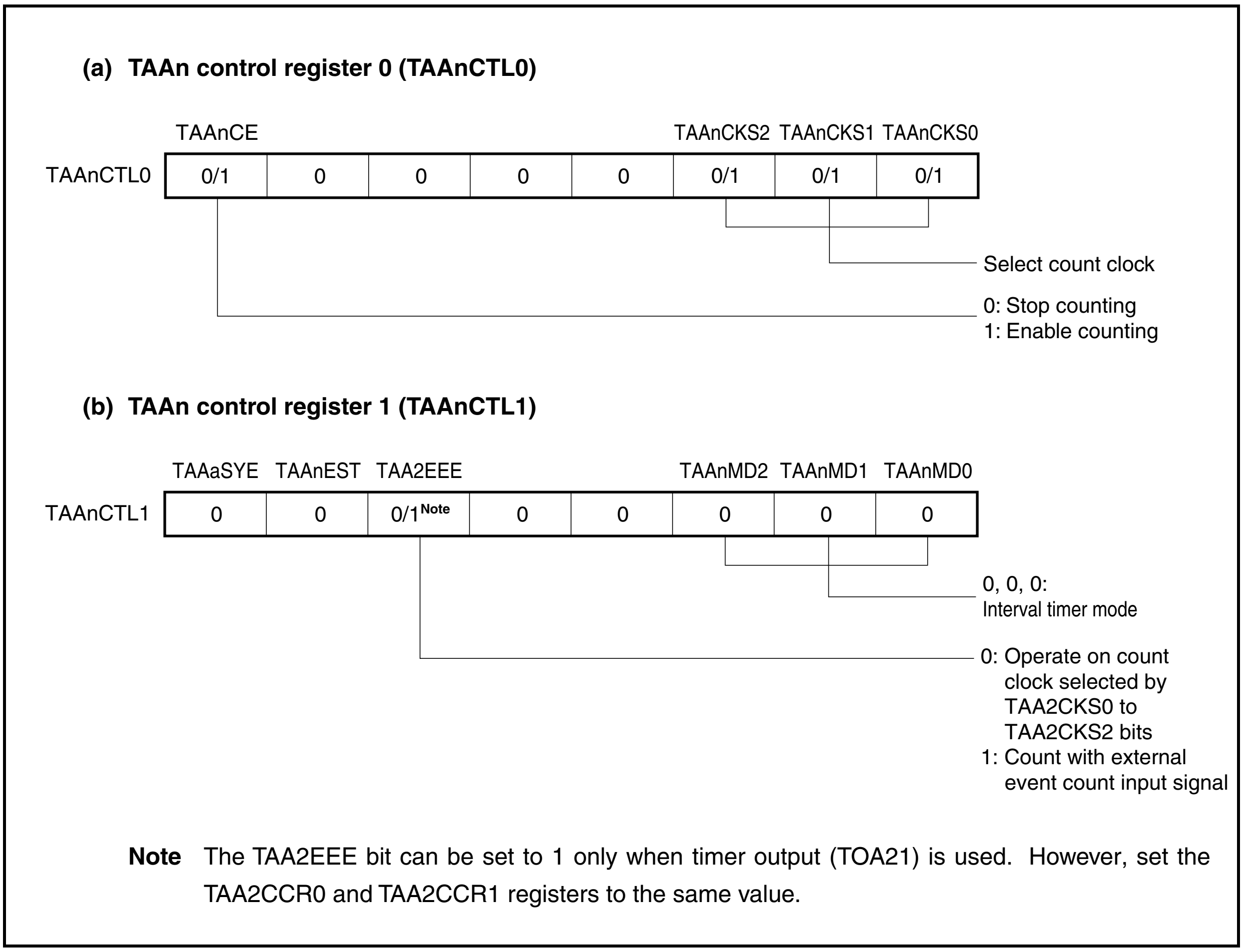
When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TOAn0 pin is inverted, and a compare match interrupt request signal (INTTAnCC0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TAAnCCR0 register + 1) × Count clock cycle

Remark n = 0 to 2

Figure 6-10. Register Setting for Interval Timer Mode Operation (1/3)



(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock
0: Stop counting
1: Enable counting

(b) TAAn control register 1 (TAAnCTL1)

	TAAaSYE	TAAnEST	TAA2EEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0	0/1Note	0	0	0	0	0

0, 0, 0:
Interval timer mode
0: Operate on count clock selected by TAA2CKS0 to TAA2CKS2 bits
1: Count with external event count input signal

Note The TAA2EEE bit can be set to 1 only when timer output (TOA21) is used. However, set the TAA2CCR0 and TAA2CCR1 registers to the same value.

Figure 6-10. Register Setting for Interval Timer Mode Operation (2/3)

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0/1	0/1	0/1	0/1

TAAnOE0:
0: Disable TOAn0 pin output
1: Enable TOAn0 pin output

TAAnOL0:
Setting of TOAn0 pin output level before count operation
0: Low level
1: High level

TAAnOE1:
0: Disable TOAn1 pin output
1: Enable TOAn1 pin output

TAAnOL1:
Setting of TOAn1 pin output level before count operation
0: Low level
1: High level

(d) TAA2 I/O control register 2 (TAA2IOC2)

					TAA2EES1	TAA2EES0	TAA2ETS1	TAA2ETS0
TAA2IOC2	0	0	0	0	0/1Note	0/1Note	0	0

TAA2EES1, TAA2EES0:
Select valid edge of external event count input (TIA20 pin).

Note The TAA2EES1 and TAA2EES0 bits can be set only when timer output (TOA21) is used. However, set the TAA2CCR0 and TAA2CCR1 registers to the same value.

(e) TAAn counter read buffer register (TAAnCNT)

By reading the TAAnCNT register, the count value of the 16-bit counter can be read.

(f) TAAn capture/compare register 0 (TAAnCCR0)

If the TAAnCCR0 register is set to D_0, the interval is as follows.

Interval = (D_0 + 1) × Count clock cycle

Figure 6-10. Register Setting for Interval Timer Mode Operation (3/3)

(g) TAAn capture/compare register 1 (TAAnCCR1)

The TAAnCCR1 register is not used in the interval timer mode. However, the set value of the TAAnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, the TOAn1 pin output is inverted and a compare match interrupt request signal (INTTAnCC1) is generated.

By setting this register to the same value as the value set in the TAAnCCR0 register, a PWM waveform with a duty factor of 50% can be output from the TOAn1 pin.

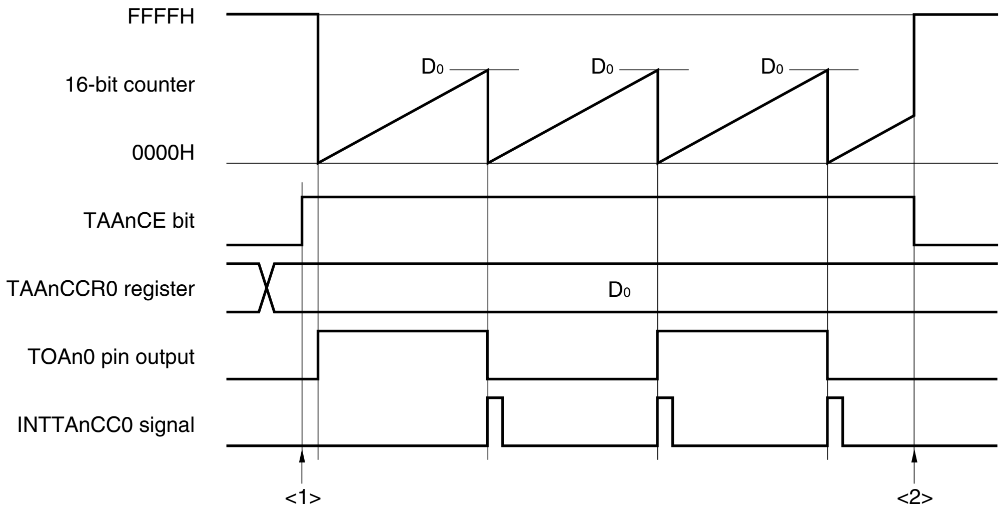
When the TAAnCCR1 register is not used, it is recommended to set the value to FFFFH. Also mask the register by the interrupt mask flag (TAAnCCIC1.TAAnCCMK1).

Remarks 1. TAA2 I/O control register 1 (TAA2IOC1) and TAAn option register 0 (TAAnOPT0) are not used in the interval timer mode.

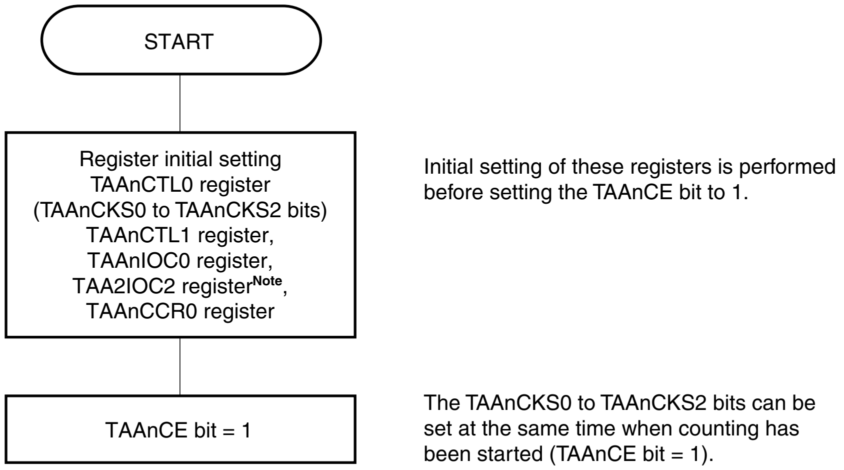
2. n = 0 to 2
a = 0, 1

(1) Interval timer mode operation flow

Figure 6-11. Software Processing Flow in Interval Timer Mode (1/2)



<1> Count operation start flow

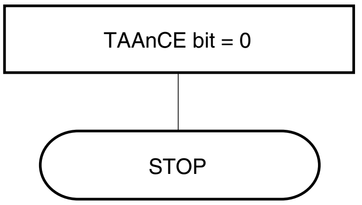

START
Register initial setting
TAAnCTL0 register
(TAAnCKS0 to TAAnCKS2 bits)
TAAnCTL1 register,
TAAnIOC0 register,
TAA2IOC2 register[Note],
TAAnCCR0 register
Initial setting of these registers is performed before setting the TAAnCE bit to 1.
TAAnCE bit = 1
The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting has been started (TAAnCE bit = 1).

Note The TAA2EES1 and TAA2EES0 bits can be set only when timer output (TOA21) is used. However, set the TAA2CCR0 and TAA2CCR1 registers to the same value.

Remark n = 0 to 2

Figure 6-11. Software Processing Flow in Interval Timer Mode (2/2)

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TAAnCE bit to 0. The output level of the TOAn0 pin is as specified by the TAAnIOC0 register.

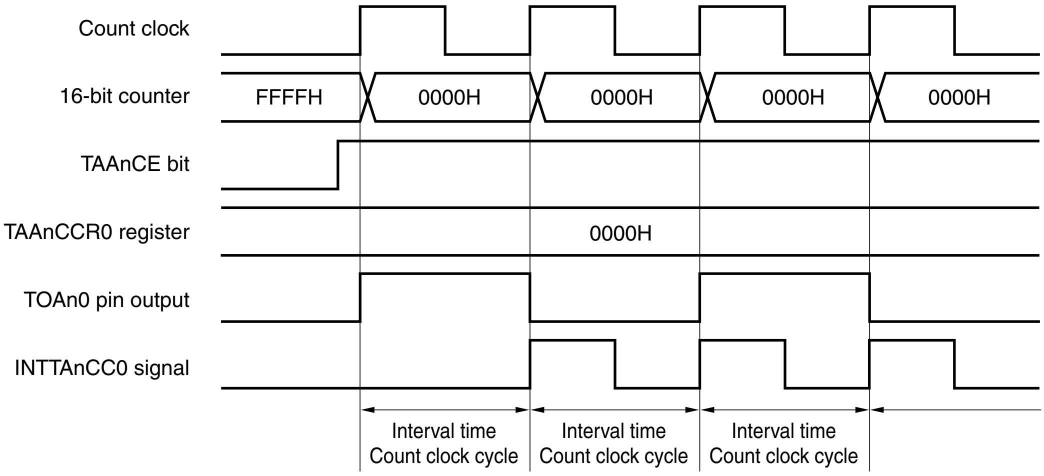
Remark n = 0 to 2

(2) Interval timer mode operation timing

(a) Operation if TAAnCCR0 register is set to 0000H

If the TAAnCCR0 register is set to 0000H, the INTTAnCC0 signal is generated at each count clock, and the output of the TOAn0 pin is inverted.

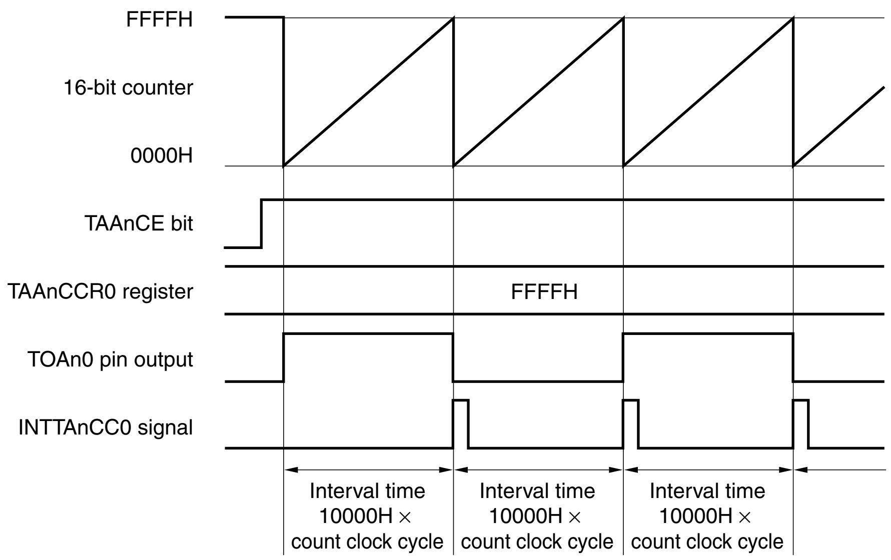
The value of the 16-bit counter is always 0000H.



Remark n = 0 to 2

(b) Operation if TAAnCCR0 register is set to FFFFH

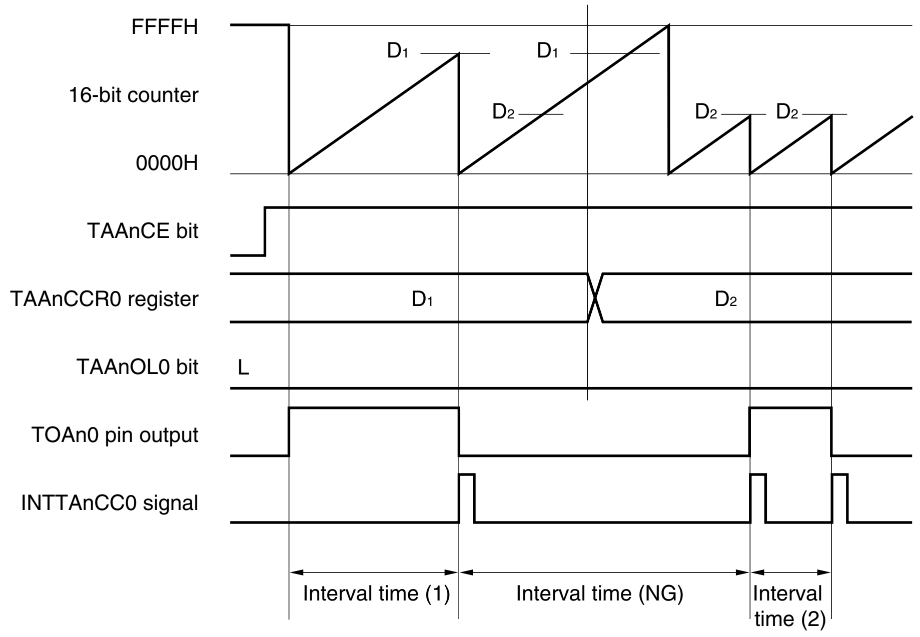
If the TAAnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTAnCC0 signal is generated and the output of the TOAn0 pin is inverted. At this time, an overflow interrupt request signal (INTTAnOV) is not generated, nor is the overflow flag (TAAnOPT0.TAAnOVF bit) set to 1.



Remark n = 0 to 2

(c) Notes on rewriting TAAnCCR0 register

If the value of the TAAnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



Remarks 1. Interval time (1): $(D_1 + 1) \times$ Count clock cycle
Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle
Interval time (2): $(D_2 + 1) \times$ Count clock cycle

2. n = 0 to 2

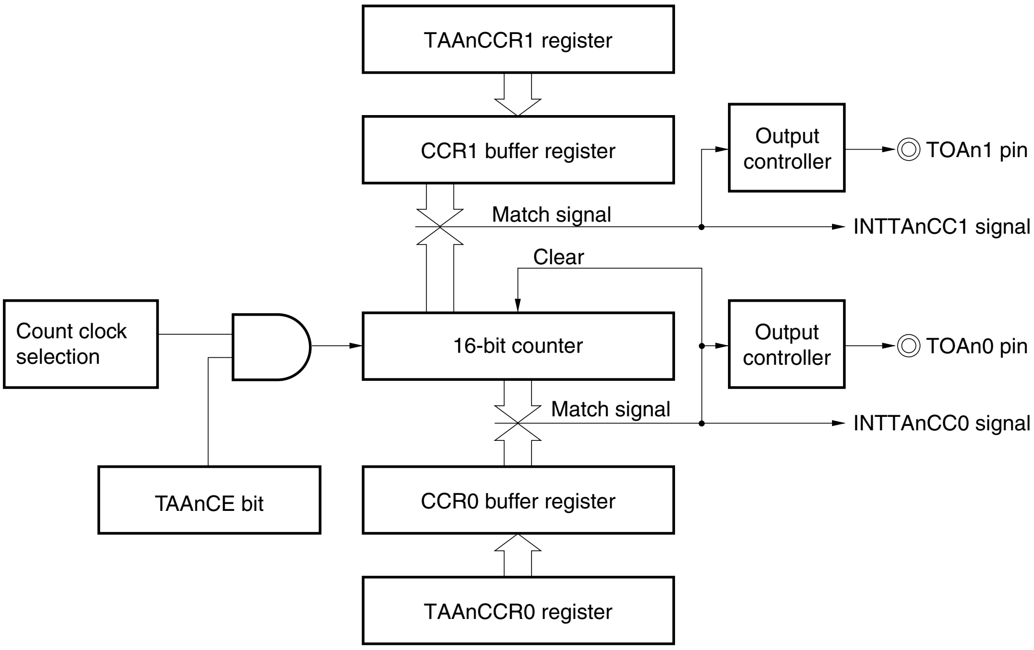
If the value of the TAAnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TAAnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTAnCC0 signal is generated and the output of the TOAn0 pin is inverted.

Therefore, the INTTAnCC0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock cycle".

(d) Operation of TAAnCCR1 register

Figure 6-12. Configuration of TAAnCCR1 Register



Remark n = 0 to 2

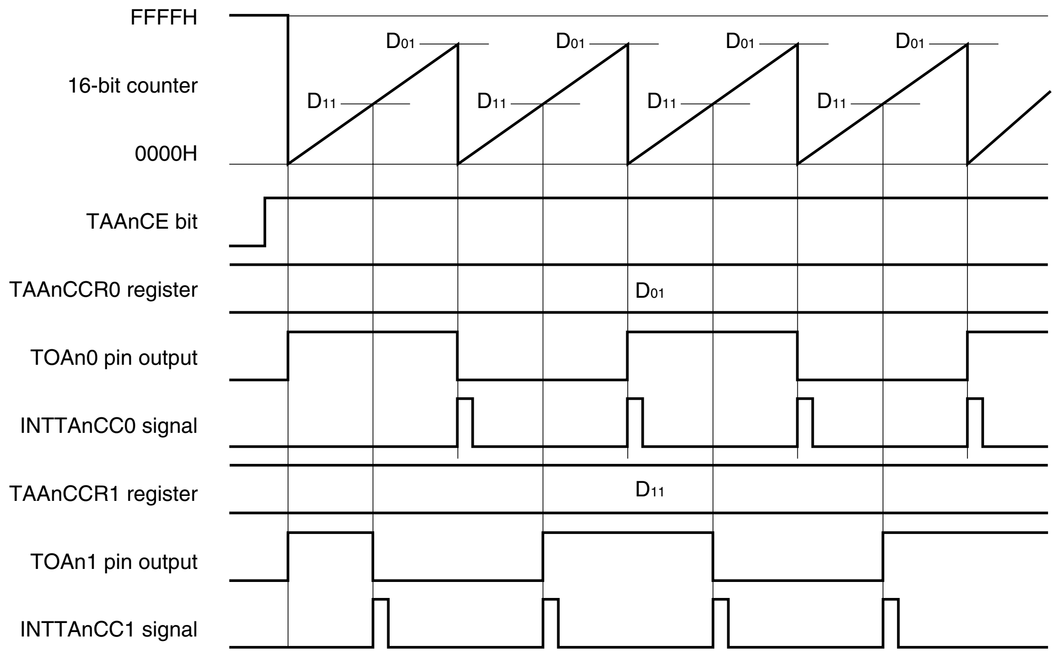
When the TAAnCCR1 register is set to the same value as the TAAnCCR0 register, the INTTAnCC0 signal is generated at the same timing as the INTTAnCC1 signal and the TOAn1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOAn1 pin.

The following shows the operation when the TAAnCCR1 register is set to other than the value set in the TAAnCCR0 register.

If the set value of the TAAnCCR1 register is less than the set value of the TAAnCCR0 register, the INTTAnCC1 signal is generated once per cycle. At the same time, the output of the TOAn1 pin is inverted.

The TOAn1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

Figure 6-13. Timing Chart When $D_{01} \geq D_{11}$

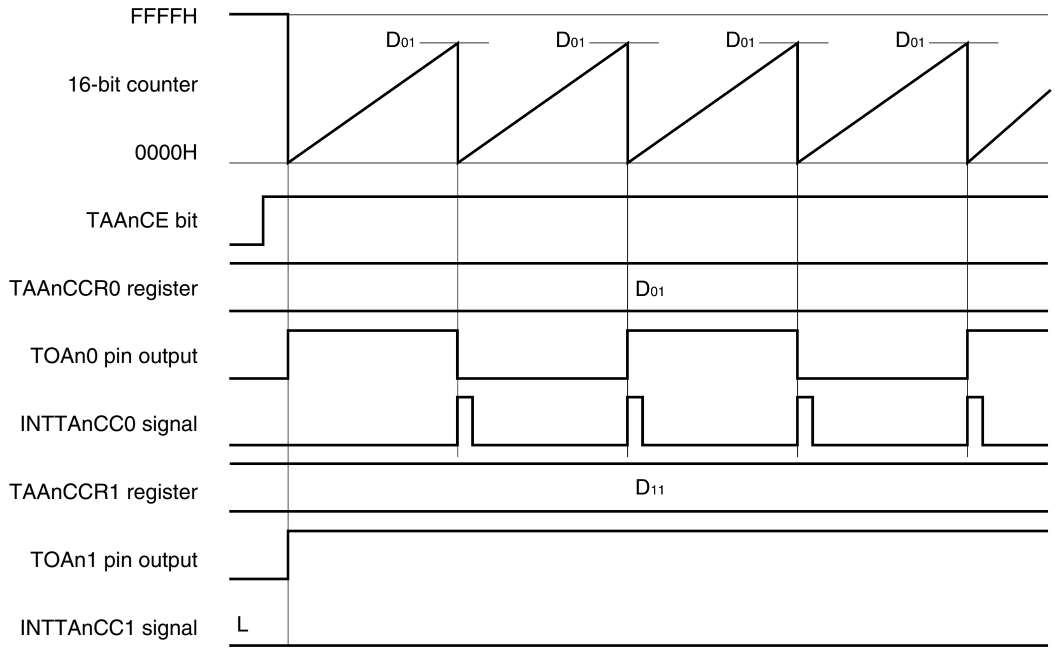


Remark n = 0 to 2

If the set value of the TAAnCCR1 register is greater than the set value of the TAAnCCR0 register, the count value of the 16-bit counter does not match the value of the TAAnCCR1 register. Consequently, the INTTAnCC1 signal is not generated, nor is the output of the TOAn1 pin changed.

When the TAAnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 6-14. Timing Chart When $D_{01} < D_{11}$



Remark n = 0 to 2

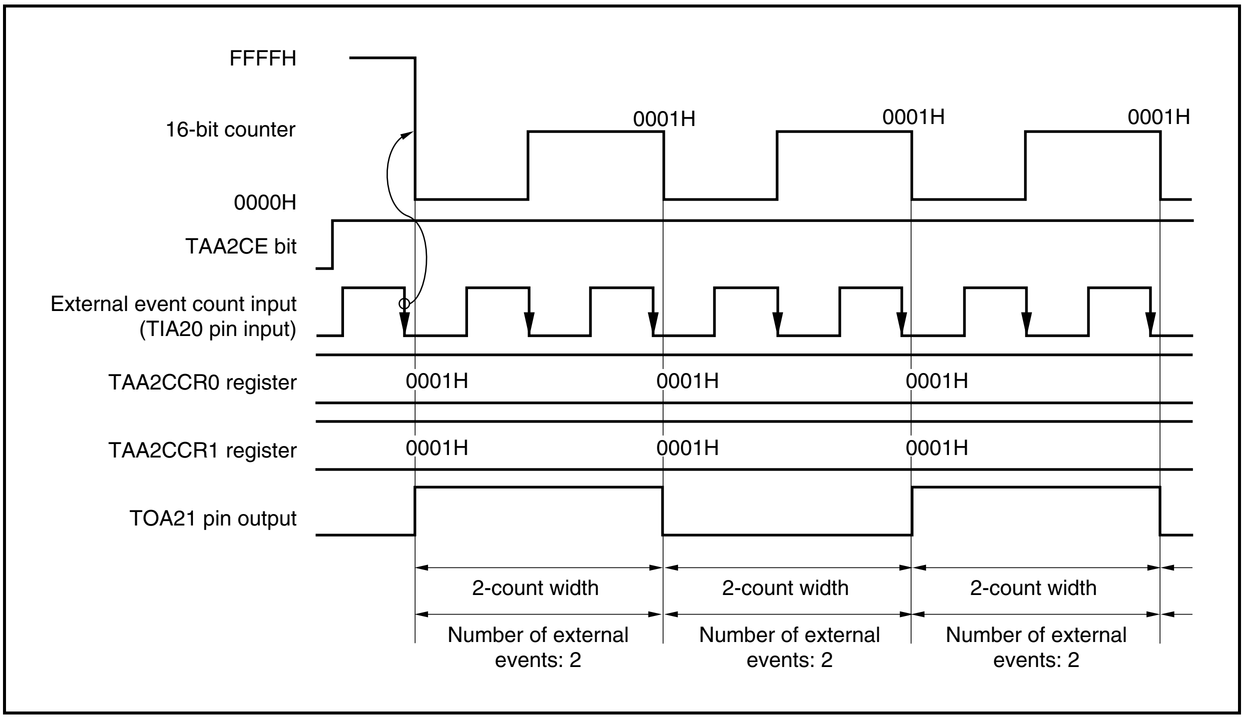
(3) Operation by external event count input (TIA20)

(a) Operation

To count the 16-bit counter at the valid edge of the external event count input (TIA20) in the interval timer mode, the 16-bit counter is cleared from FFFFH to 0000H by the valid edge of the external event count after the TAA2CE bit is set from 0 to 1.

When 0001H is set to both the TAA2CCR0 and TAA2CCR1 registers, the TOA21 pin output is inverted each time the 16-bit counter counts twice.

The TAA2CTL1.TAA2EEE bit can be set to 1 in the interval timer mode only when the timer output (TOA21) is used with the external event count input.



6.6.2 External event count mode (TAA2MD2 to TAA2MD0 bits = 001)

This mode is valid only in TAA2.

In the external event count mode, the valid edge of the external event count input (TIA20) is counted when the TAA2CTL0.TAA2CE bit is set to 1, and an interrupt request signal (INTTA2CC0) is generated each time the number of edges set by the TAA2CCR0 register have been counted. The TOA20 and TOA21 pins cannot be used. When using the TOA21 pin for external event count input, set the TAA2CTL1.TAA2EEE bit to 1 in the interval timer mode (see **6.6.1 (3) Operation by external event count input (TIA20)**).

The TAA2CCR1 register is not used in the external event count mode.

Figure 6-15. Configuration in External Event Count Mode

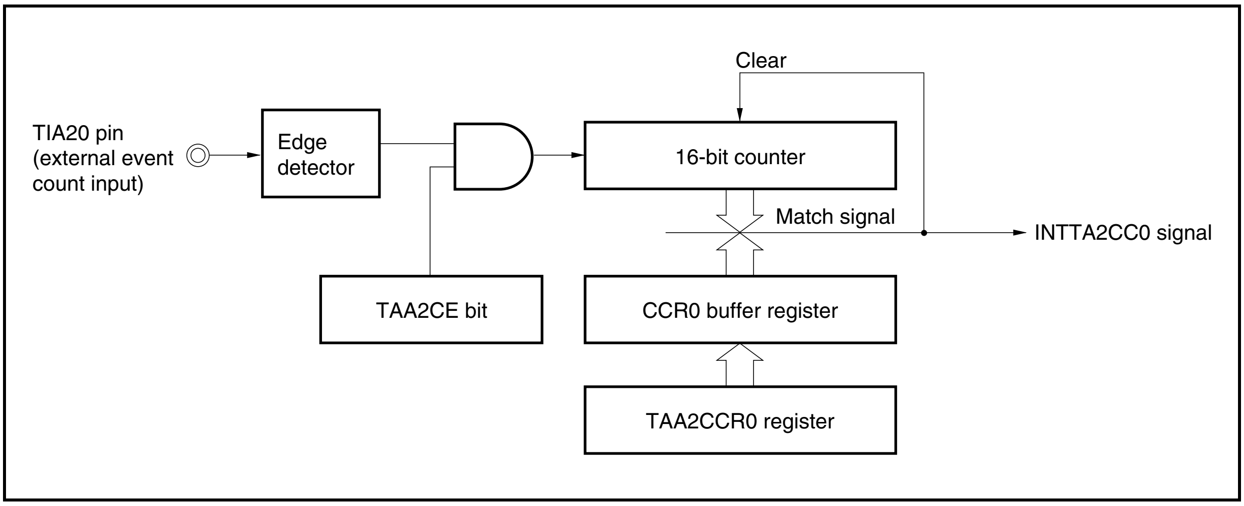
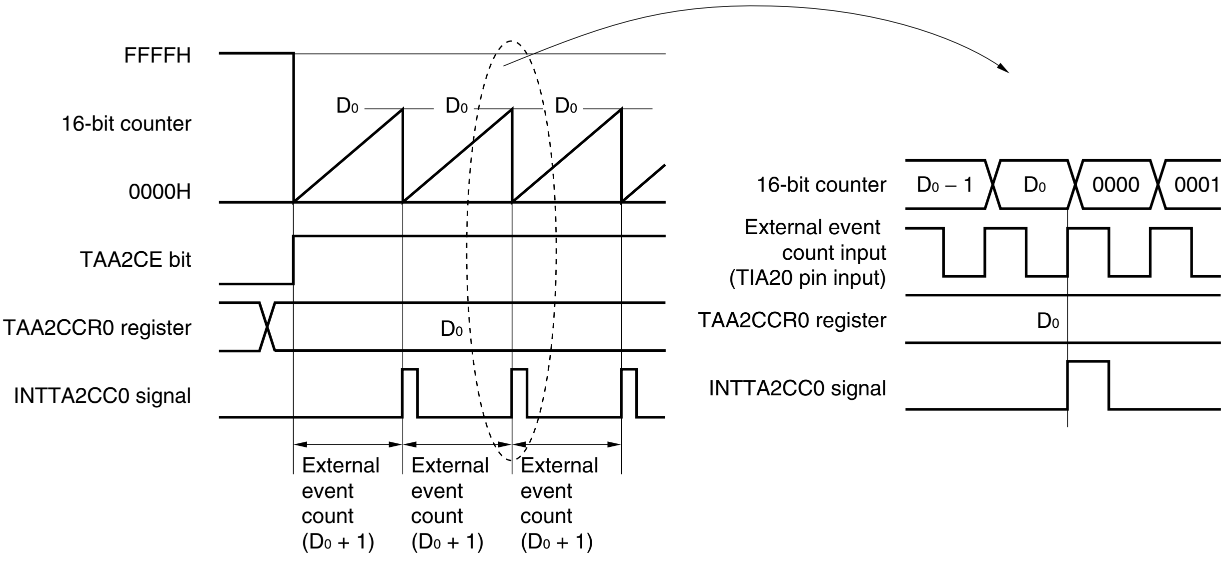


Figure 6-16. Basic Timing in External Event Count Mode



Remark This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

When the TAA2CE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TAA2CCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTA2CC0) is generated.

The INTTA2CC0 signal is generated each time the valid edge of the external event count input has been detected “value set to TAA2CCR0 register + 1” times.

Figure 6-17. Register Setting for Operation in External Event Count Mode (1/2)

(a) TAA2 control register 0 (TAA2CTL0)

	TAA2CE					TAA2CKS2	TAA2CKS1	TAA2CKS0
TAA2CTL0	0/1	0	0	0	0	0	0	0

TAA2CE:
0: Stop counting
1: Enable counting

(b) TAA2 control register 1 (TAA2CTL1)

	TAAaSYE	TAA2EST	TAA2EEE			TAA2MD2	TAA2MD1	TAA2MD0
TAA2CTL1	0	0	0	0	0	0	0	1

TAA2MD2 to TAA2MD0:
0, 0, 1:
External event count mode

(c) TAA2 I/O control register 2 (TAA2IOC2)

					TAA2EES1	TAA2EES0	TAA2ETS1	TAA2ETS0
TAA2IOC2	0	0	0	0	0/1	0/1	0	0

TAA2EES1, TAA2EES0:
Select valid edge
of external event
count input (TIA20 pin)

(d) TAA2 counter read buffer register (TAA2CNT)

The count value of the 16-bit counter can be read by reading the TAA2CNT register.

(e) TAA2 capture/compare register 0 (TAA2CCR0)

If the TAA2CCR0 register is set to D_0, the count is cleared when the number of external events has reached (D_0 + 1) and the compare match interrupt request signal (INTTA2CC0) is generated.

Figure 6-17. Register Setting for Operation in External Event Count Mode (2/2)

(f) TAA2 capture/compare register 1 (TAA2CCR1)

The TAA2CCR1 register is not used in the external event count mode. However, the set value of the TAA2CCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTA2CC1) is generated.

When the TAA2CCR1 register is not used, it is recommended to set the value to FFFFH. Also mask the register by the interrupt mask flag (TAA2CCIC1.TAA2CCMK1).

Cautions 1. Set the TAA2IOC0 register to 00H.

2. When an external clock is used as the count clock, the external clock can be input only from the TIA20 pin.

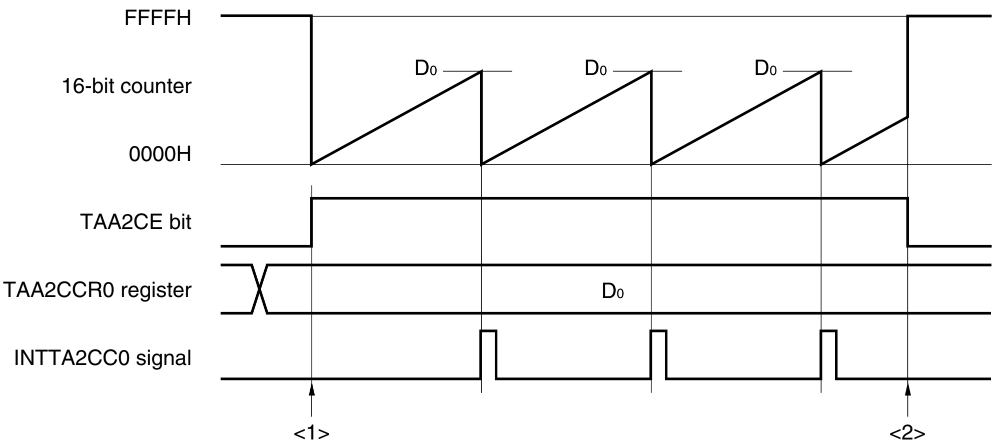
At this time, set the TAA2IOC1.TAA2IS1 and TAA2IOC1.TAA2IS0 bits to 00 (capture trigger input (TIA20 pin): no edge detection)

Remarks 1. TAA2 I/O control register 1 (TAA2IOC1) and TAA2 option register 0 (TAA2OPT0) are not used in the external event count mode.

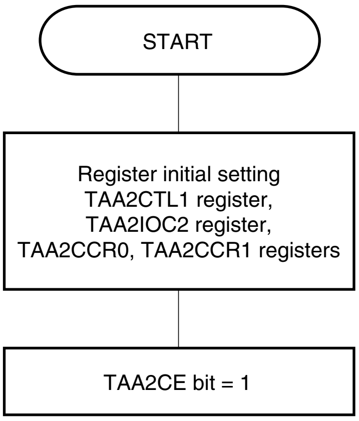
2. a = 0, 1

(1) External event count mode operation flow

Figure 6-18. Software Processing Flow in External Event Count Mode

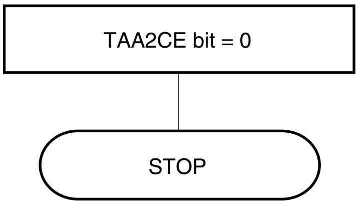


<1> Count operation start flow



Initial setting of these registers
is performed before setting the
TAA2CE bit to 1.

<2> Count operation stop flow



The counter is initialized and counting
is stopped by clearing the TAA2CE bit to 0.

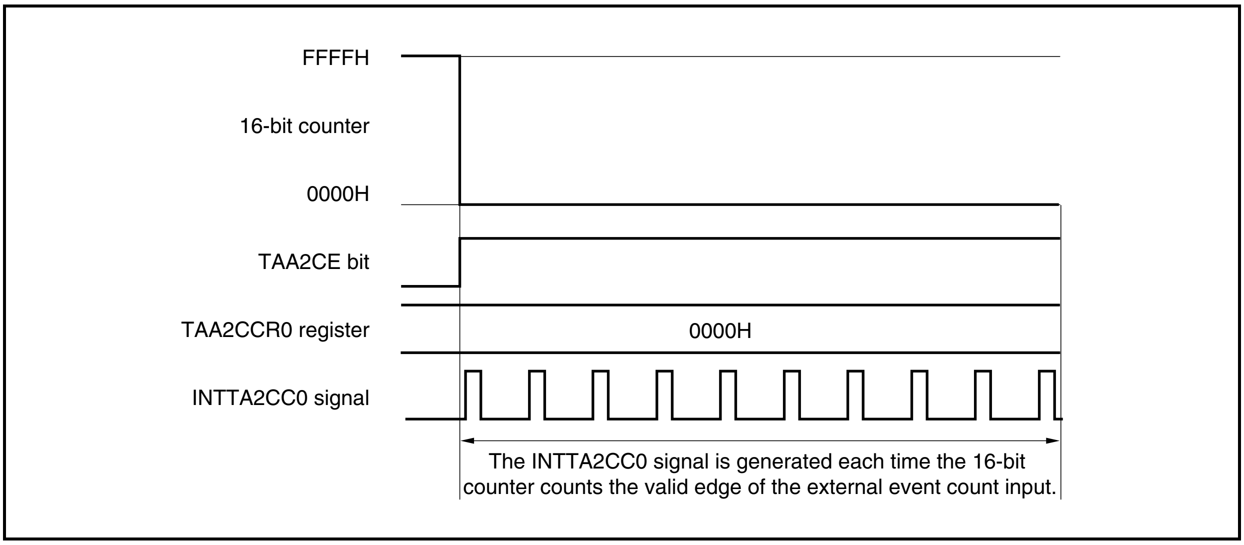
(2) Operation timing in external event count mode

Caution In the external event count mode, use of the timer output (TOA20, TOA21) is disabled. If using timer output (TOA21) with external event count input (TIA20), set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TAA2CTL1.TAA2EEE bit = 1) (see 6.6.1 (3) Operation by external event count input (TIA20)).

(a) Operation if TAA2CCR0 register is set to 0000H

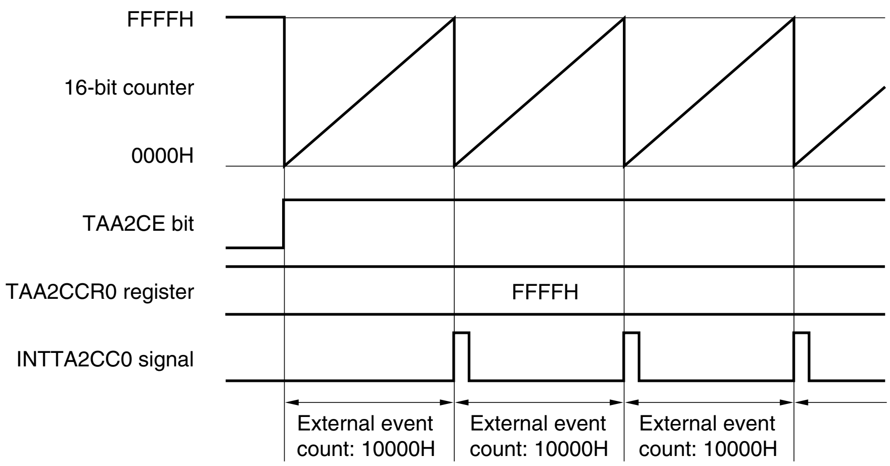
When the TAA2CCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates the INTTA2CC0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

The value of the 16-bit counter is always 0000H.


FFFFH
16-bit counter
0000H
TAA2CE bit
TAA2CCR0 register
0000H
INTTA2CC0 signal
The INTTA2CC0 signal is generated each time the 16-bit counter counts the valid edge of the external event count input.

(b) Operation if TAA2CCR0 register is set to FFFFH

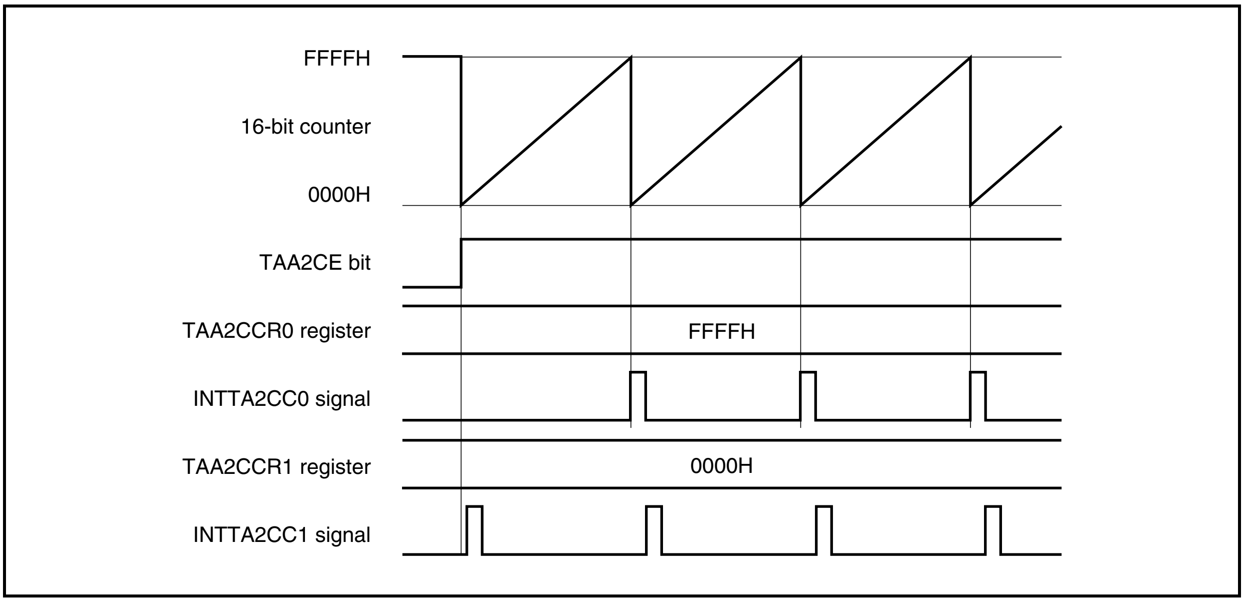
If the TAA2CCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTA2CC0 signal is generated. At this time, the TAA2OPT0.TAA2OVF bit is not set.


FFFFH
16-bit counter
0000H
TAA2CE bit
TAA2CCR0 register
FFFFH
INTTA2CC0 signal
External event count: 10000H
External event count: 10000H
External event count: 10000H

(c) Operation with TAA2CCR0 register set to FFFFH and TAA2CCR1 register to 0000H

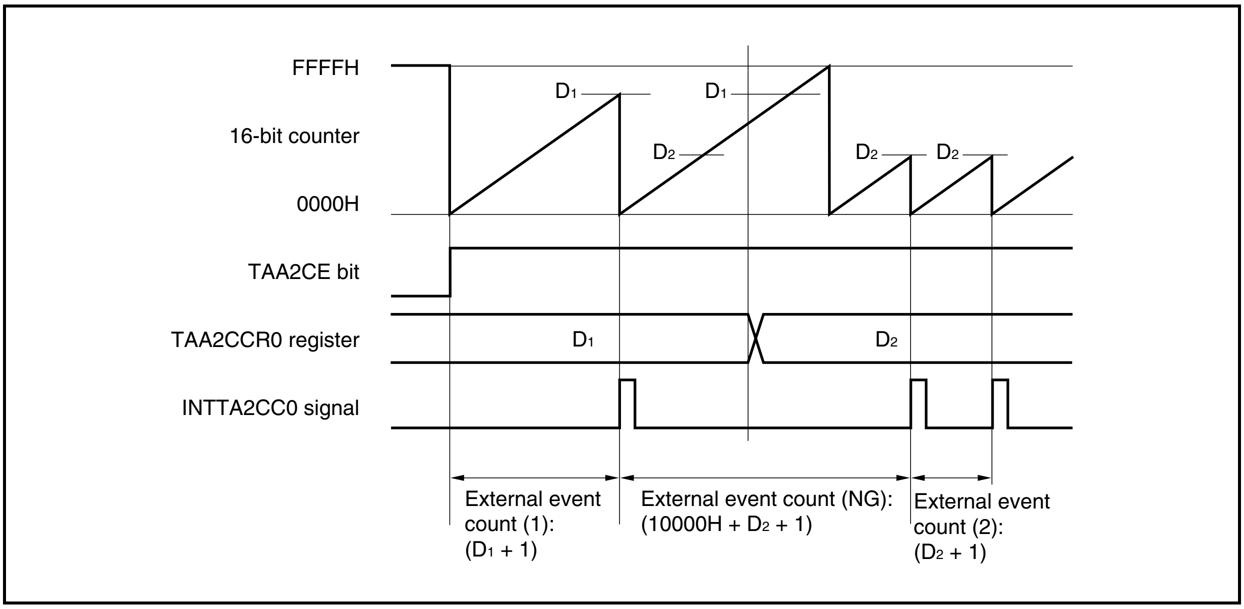
When the TAA2CCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTA2CC0 signal is generated. At this time, the TAA2OPT0.TAA2OVF bit is not set.

If the TAA2CCR1 register is set to 0000H, the INTTA2CC1 signal is generated when the 16-bit counter is cleared to 0000H.



(d) Notes on rewriting the TAA2CCR0 register

If the value of the TAA2CCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.


FFFFH
16-bit counter
0000H
TAA2CE bit
TAA2CCR0 register
INTTA2CC0 signal
D1
D1
D2
D2
D2
D1
D2
External event count (1): (D1 + 1)
External event count (NG): (10000H + D2 + 1)
External event count (2): (D2 + 1)

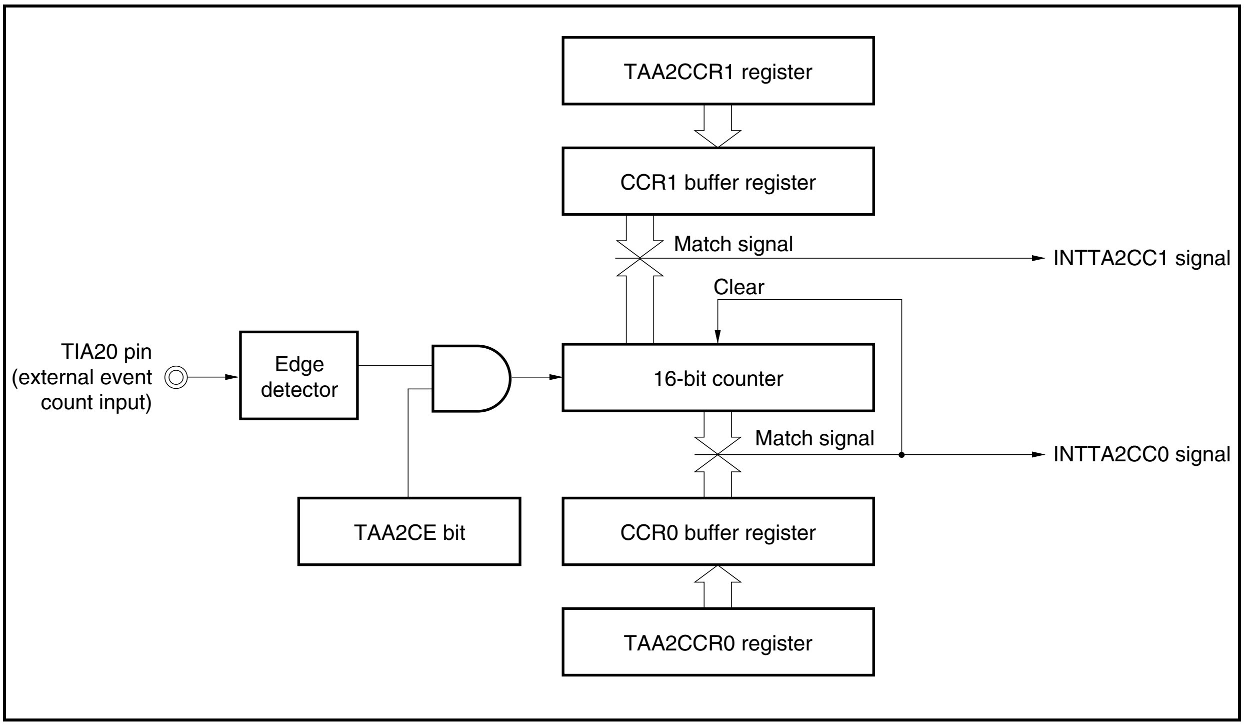
If the value of the TAA2CCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TAA2CCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTA2CC0 signal is generated.

Therefore, the INTTA2CC0 signal may not be generated at the valid edge count of "(D_1 + 1) times" or "(D_2 + 1) times" originally expected, but may be generated at the valid edge count of "(10000H + D_2 + 1) times".

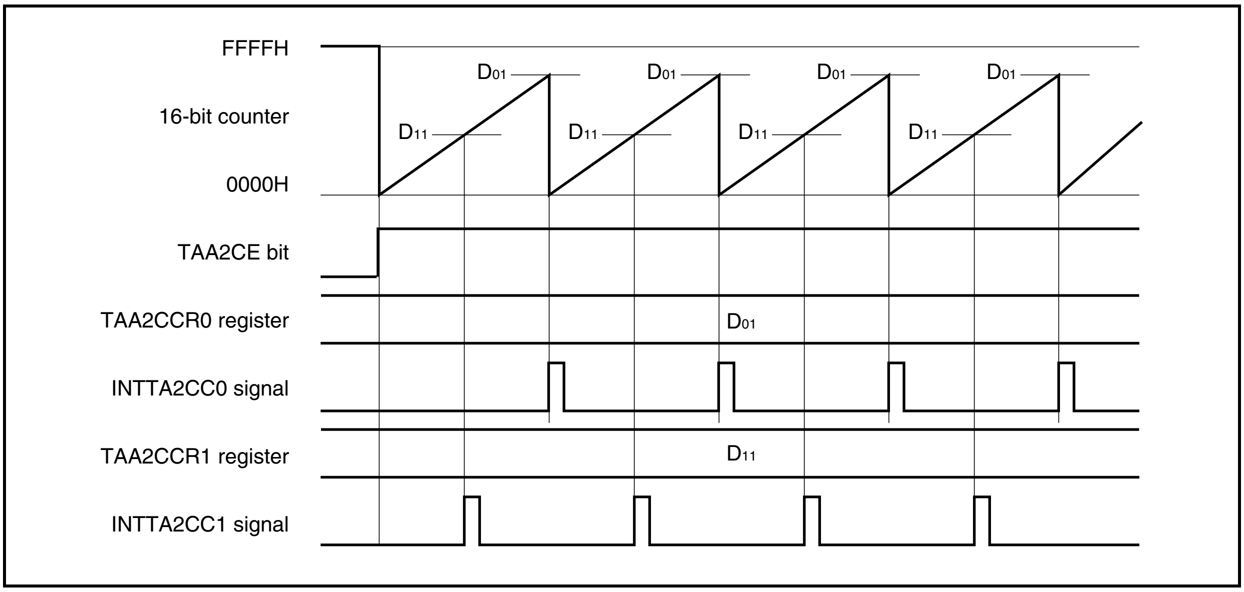
(e) Operation of TAA2CCR1 register

Figure 6-19. Configuration of TAA2CCR1 Register



If the set value of the TAA2CCR1 register is smaller than the set value of the TAA2CCR0 register, the INTTA2CC1 signal is generated once per cycle.

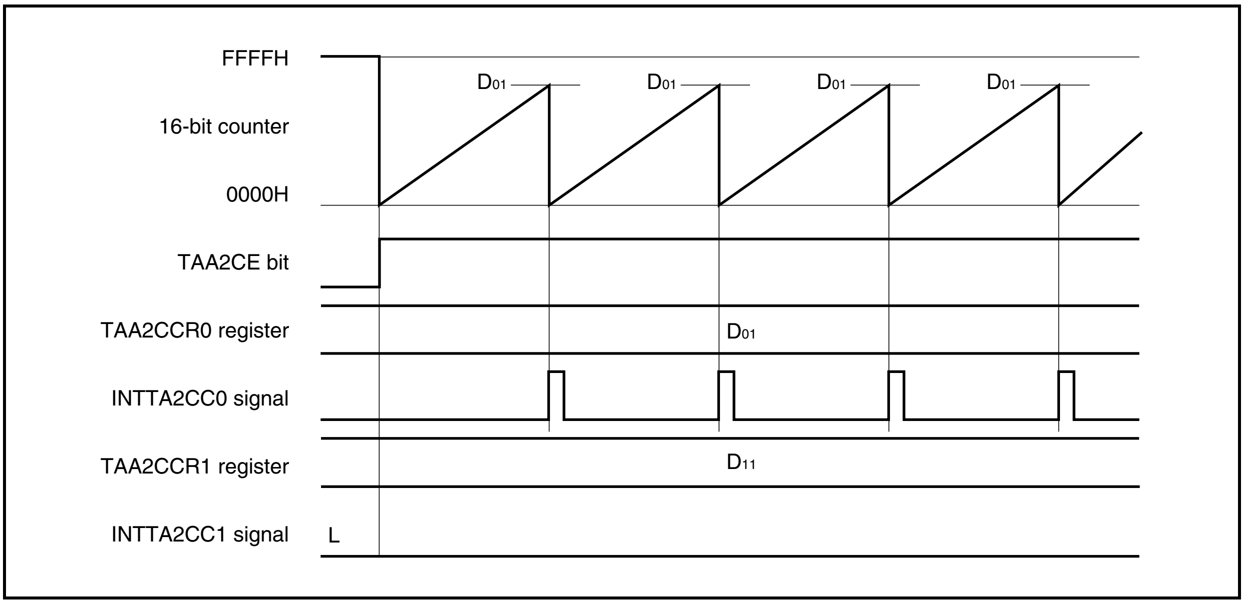
Figure 6-20. Timing Chart When $D_{01} \geq D_{11}$



If the set value of the TAA2CCR1 register is greater than the set value of the TAA2CCR0 register, the INTTA2CC1 signal is not generated because the count value of the 16-bit counter and the value of the TAA2CCR1 register do not match.

When the TAA2CCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 6-21. Timing Chart When D_{01} < D_{11}



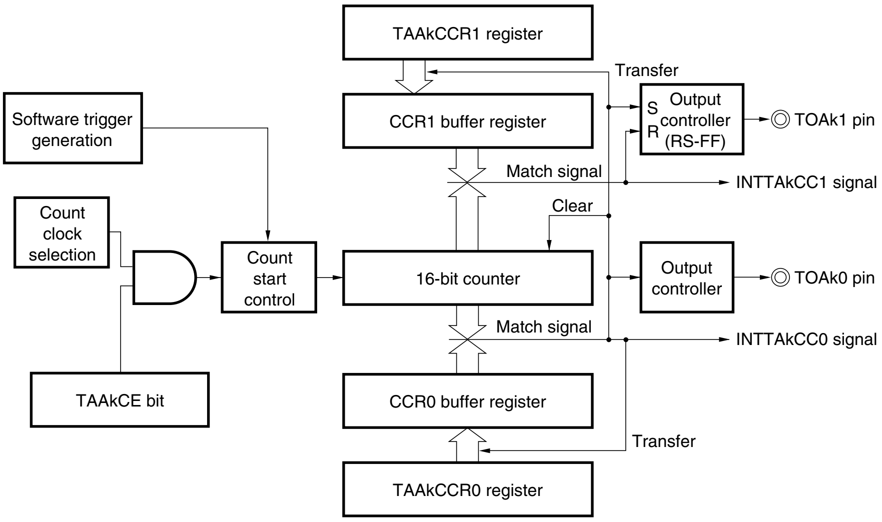
6.6.3 External trigger pulse output mode (TAAnMD2 to TAAnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter AA waits for a trigger when the TAAnCTL0.TAAnCE bit is set to 1.

For TAA0 and TAA1, the counter starts incrementing when a software trigger is detected and a PWM waveform is output from the TOAk1 pin. A PWM waveform with 50% duty and that has the value of the TAAkCCR0 register + 1 as half its cycle can also be output from the TOAk0 pin.

For TAA2, the counter starts incrementing when the valid edge of the external trigger input (TIA20) is detected and a PWM waveform is output from the TOA21 pin. Pulses can also be output by generating a software trigger instead of using the external trigger input. When using a software trigger, a PWM waveform with 50% duty and that has the value of the TAA2CCR0 register + 1 as half its cycle can also be output from the TOA20 pin.

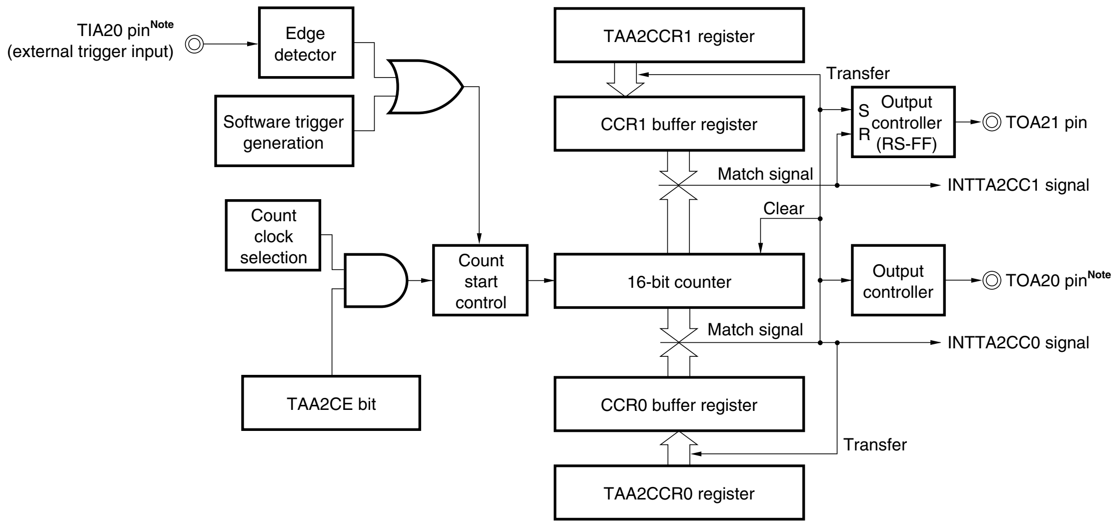
Figure 6-22. TAA0 and TAA1 Configuration in External Trigger Pulse Output Mode



Caution **In the external trigger pulse output mode, select the internal clock as the count clock (by clearing the TAAkCTL1.TAAkEEE bit to 0).**

Remark k = 0, 1

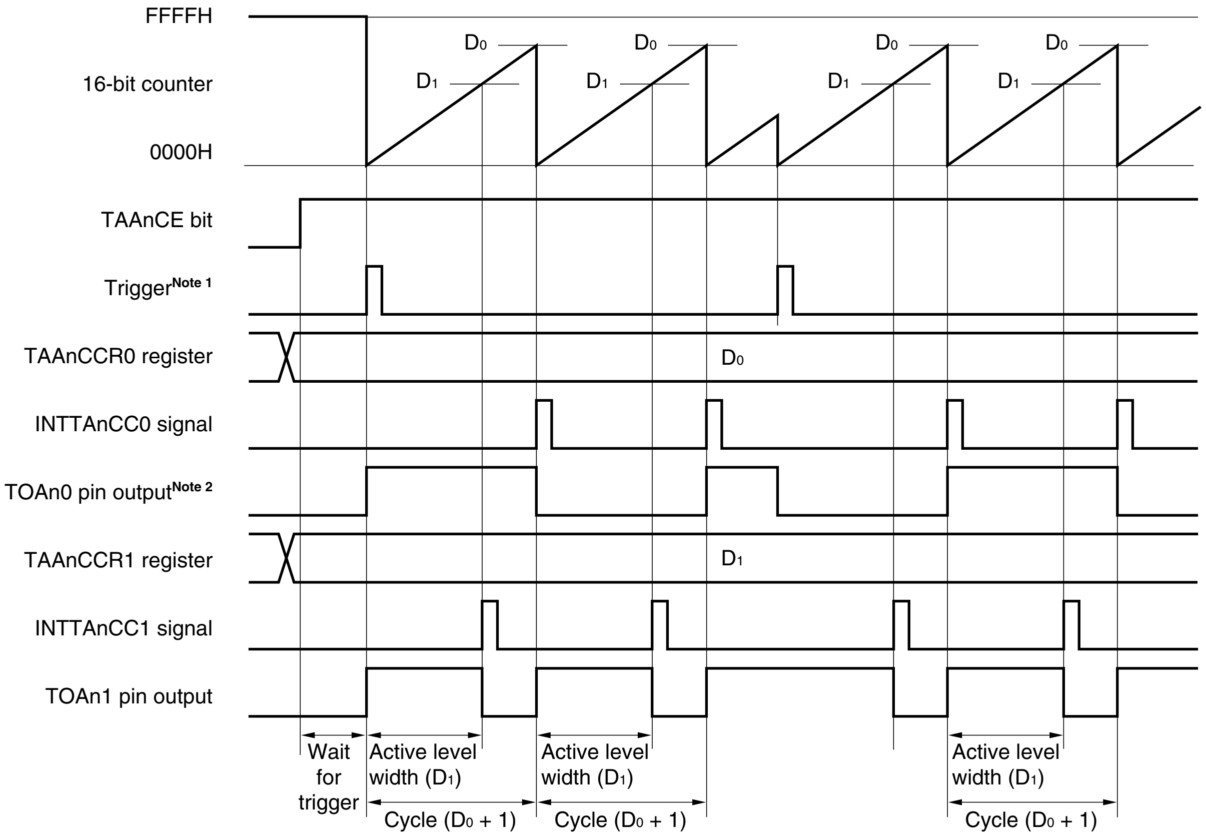
Figure 6-23. Configuration of TAA2 in External Trigger Pulse Output Mode



Note Because the external trigger input pin (TIA20) and timer output pin (TOA20) are the same pin, the two functions cannot be used at the same time.

Caution In the external trigger pulse output mode, select the internal clock as the count clock (by clearing the TAA2CTL1.TAA2EEE bit to 0).

Figure 6-24. Basic Timing in External Trigger Pulse Output Mode



Notes 1. A software trigger for TAA0 and TAA1 or an external trigger input (from the TIA20 pin) for TAA2

2. For TAA2, this function can only be used by using a software trigger.

16-bit timer/event counter AA waits for a trigger when the TAAnCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TOAn1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOAn0 pin is inverted. The TOAn1 pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TAAnCCR1 register) × Count clock cycle
Cycle = (Set value of TAAnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TAAnCCR1 register)/(Set value of TAAnCCR0 register + 1)

The compare match interrupt request signal INTTAnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTAnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TAAnCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input (TIA20), or setting the software trigger (TAAnCTL1.TAAnEST bit) to 1 is used as the trigger.

Remark n = 0 to 2
a = 0, 1

Figure 6-25. Setting of Registers in External Trigger Pulse Output Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TAAnCKS2 to TAAnCKS0: Select count clock

TAAnCE:
0: Stop counting
1: Enable counting

(b) TAAn control register 1 (TAAnCTL1)

	TAAaSYE	TAAnEST	TAA2EEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0/1	0	0	0	0	1	0

TAAnMD2 to TAAnMD0:
0, 1, 0:
External trigger pulse output mode

TAA2EEE:
0: Operate on count clock selected by TAA2CKS0 to TAA2CKS2 bits

TAAnEST: Generate software trigger when 1 is written

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

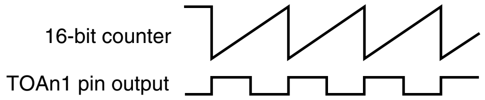
TAAnOE0:
0: Disable TOAn0 pin output
1: Enable TOAn0 pin output

TAAnOL0:
Setting of TOAn0 pin output level while waiting for external trigger
0: Low level
1: High level

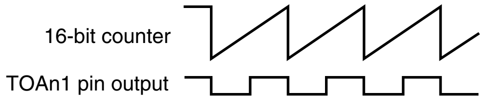
TAAnOE1:
0: Disable TOAn1 pin output
1: Enable TOAn1 pin output

TAAnOL1:
Setting of TOAn1 pin output level while waiting for external trigger
0: Low level
1: High level

- When TAAnOL1 bit = 0



- When TAAnOL1 bit = 1



Note Clear this bit to 0 when the TOAn0 pin is not used in the external trigger pulse output mode.

Figure 6-25. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TAA2 I/O control register 2 (TAA2IOC2)

					TAA2EES1	TAA2EES0	TAA2ETS1	TAA2ETS0
TAA2IOC2	0	0	0	0	0	0	0/1	0/1

Select valid edge of external trigger input (TIA20 pin)

(e) TAAn counter read buffer register (TAAnCNT)

The value of the 16-bit counter can be read by reading the TAAnCNT register.

(f) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

If D_0 is set to the TAAnCCR0 register and D_1 to the TAAnCCR1 register, the cycle and active level of the PWM waveform are as follows.

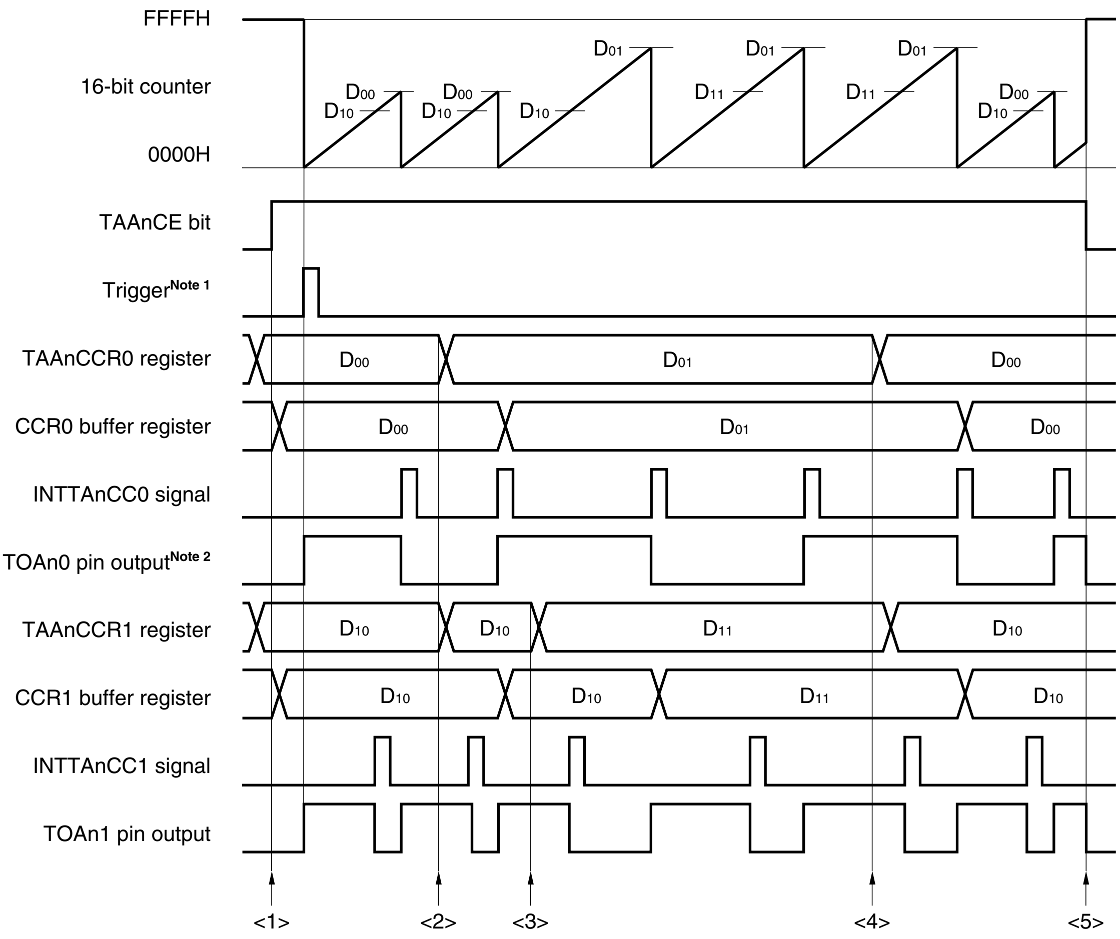
Cycle = $(D_0 + 1)$ × Count clock cycle
Active level width = D_1 × Count clock cycle

Remarks **1.** TAA2 I/O control register 1 (TAA2IOC1) and TAAn option register 0 (TAAnOPT0) are not used in the external trigger pulse output mode.

2. n = 0 to 2

(1) Operation flow in external trigger pulse output mode

Figure 6-26. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

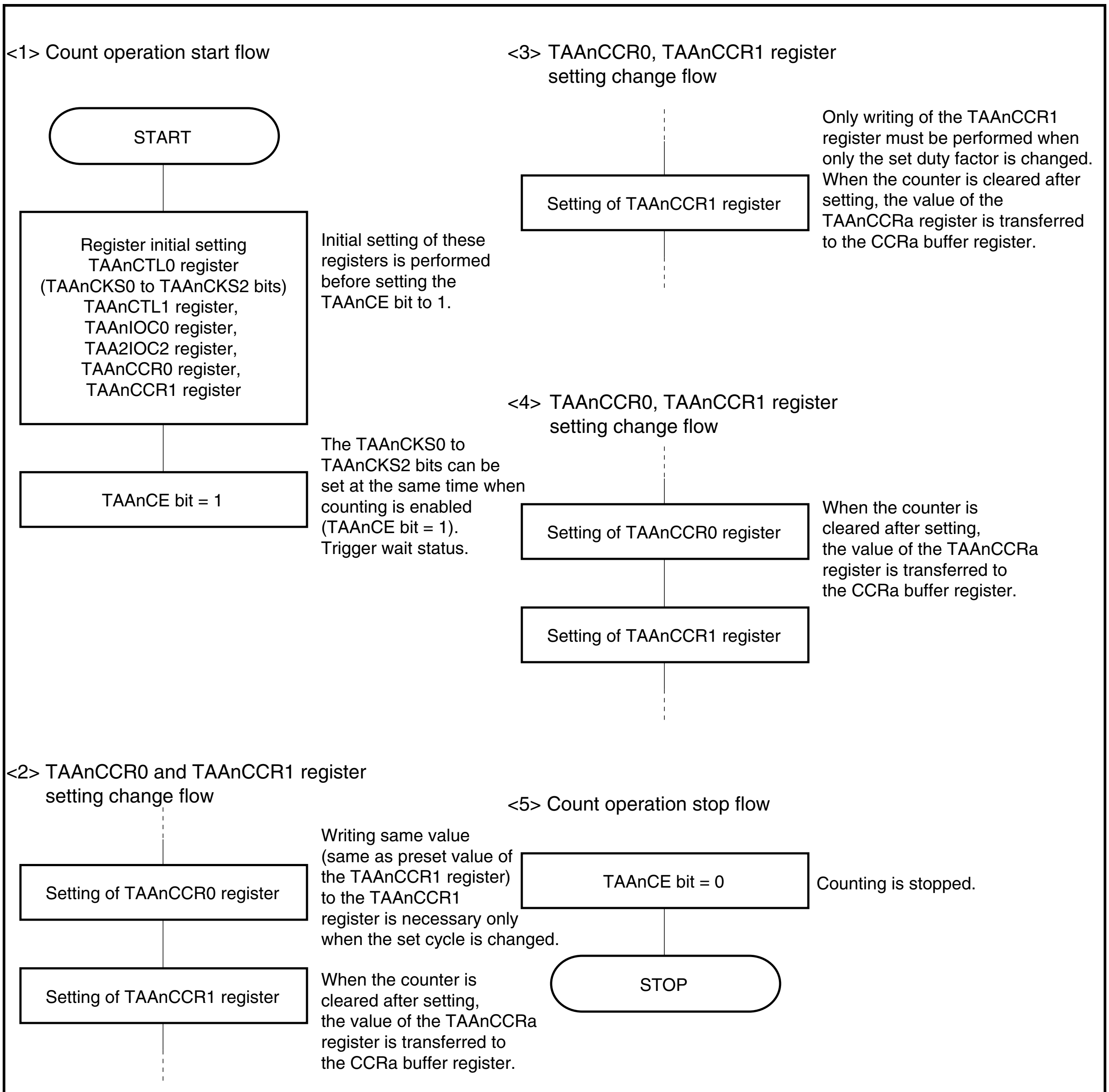


Notes 1. A software trigger for TAA0 and TAA1 or an external trigger input (from the TIA20 pin) for TAA2

2. For TAA2, this function can only be used by using a software trigger.

Remark n = 0 to 2

Figure 6-26. Software Processing Flow in External Trigger Pulse Output Mode (2/2)


START
Register initial setting
TAAnCTL0 register
(TAAnCKS0 to TAAnCKS2 bits)
TAAnCTL1 register,
TAAnIOC0 register,
TAA2IOC2 register,
TAAnCCR0 register,
TAAnCCR1 register
TAAnCE bit = 1
Setting of TAAnCCR0 register
Setting of TAAnCCR1 register
Setting of TAAnCCR1 register
Setting of TAAnCCR0 register
Setting of TAAnCCR1 register
TAAnCE bit = 0
STOP

<1> Count operation start flow

Initial setting of these registers is performed before setting the TAAnCE bit to 1.

The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting is enabled (TAAnCE bit = 1). Trigger wait status.

<2> TAAnCCR0 and TAAnCCR1 register setting change flow

Writing same value (same as preset value of the TAAnCCR1 register) to the TAAnCCR1 register is necessary only when the set cycle is changed.

When the counter is cleared after setting, the value of the TAAnCCRa register is transferred to the CCRa buffer register.

<3> TAAnCCR0, TAAnCCR1 register setting change flow

Only writing of the TAAnCCR1 register must be performed when only the set duty factor is changed. When the counter is cleared after setting, the value of the TAAnCCRa register is transferred to the CCRa buffer register.

<4> TAAnCCR0, TAAnCCR1 register setting change flow

When the counter is cleared after setting, the value of the TAAnCCRa register is transferred to the CCRa buffer register.

<5> Count operation stop flow

Counting is stopped.

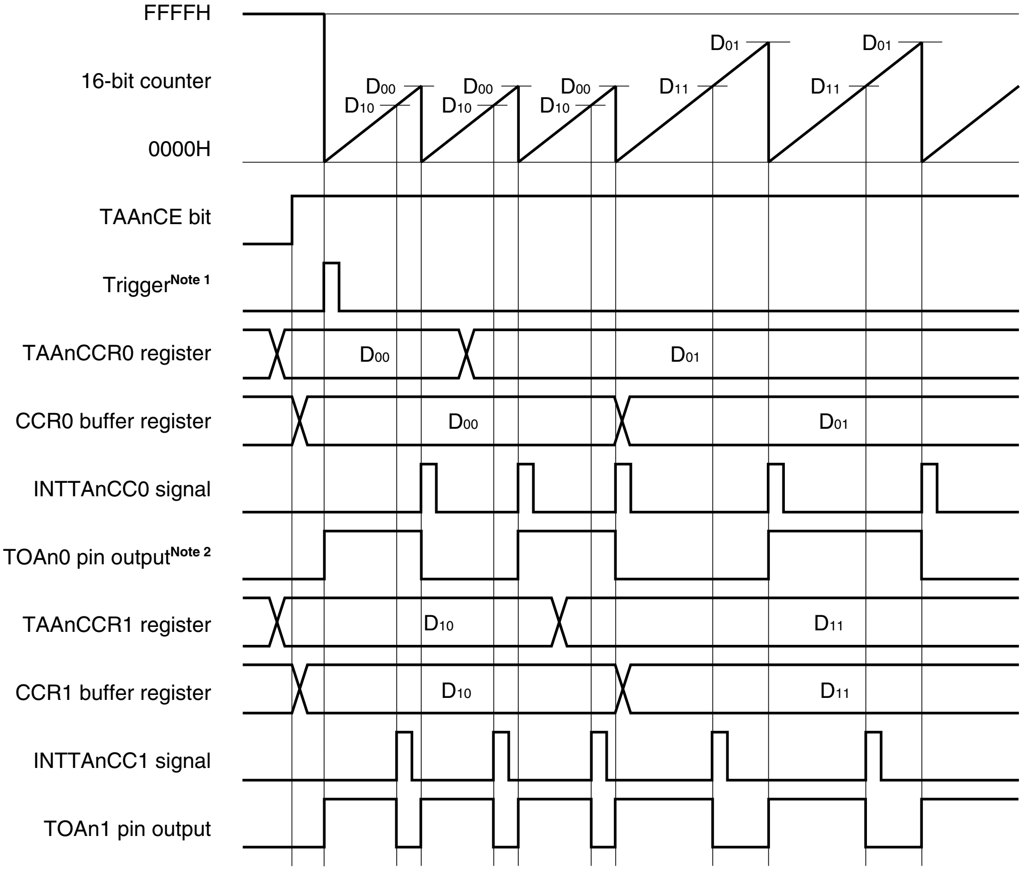
Remark n = 0 to 2
a = 0, 1

(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TAAnCCR1 register last.

Rewrite the TAAnCCRa register after writing the TAAnCCR1 register after the INTTAnCC0 signal is detected.



Notes 1. A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

2. For TAA2, this function can only be used by using a software trigger.

Remark n = 0 to 2

In order to transfer data from the TAAnCCRa register to the CCRa buffer register, the TAAnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TAAnCCR0 register and then set the active level width to the TAAnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TAAnCCR0 register, and then write the same value (same as preset value of the TAAnCCR1 register) to the TAAnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TAAnCCR1 register has to be set.

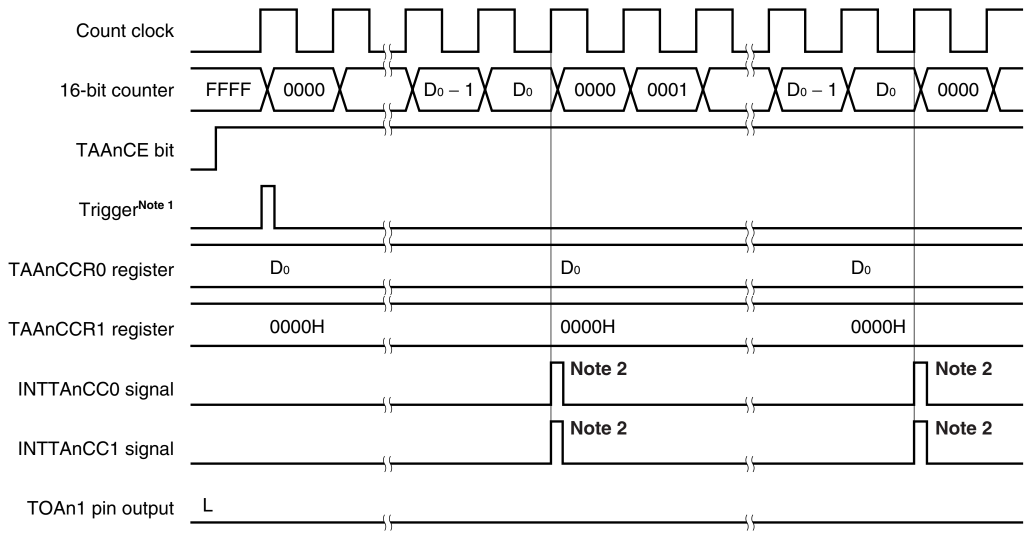
After data is written to the TAAnCCR1 register, the value written to the TAAnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TAAnCCR0 or TAAnCCR1 register again after writing the TAAnCCR1 register once, do so after the INTTAnCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TAAnCCRa register to the CCRa buffer register conflicts with writing the TAAnCCRa register.

Remark n = 0 to 2
a = 0, 1

(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TAAnCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTAnCC0 and INTTAnCC1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.

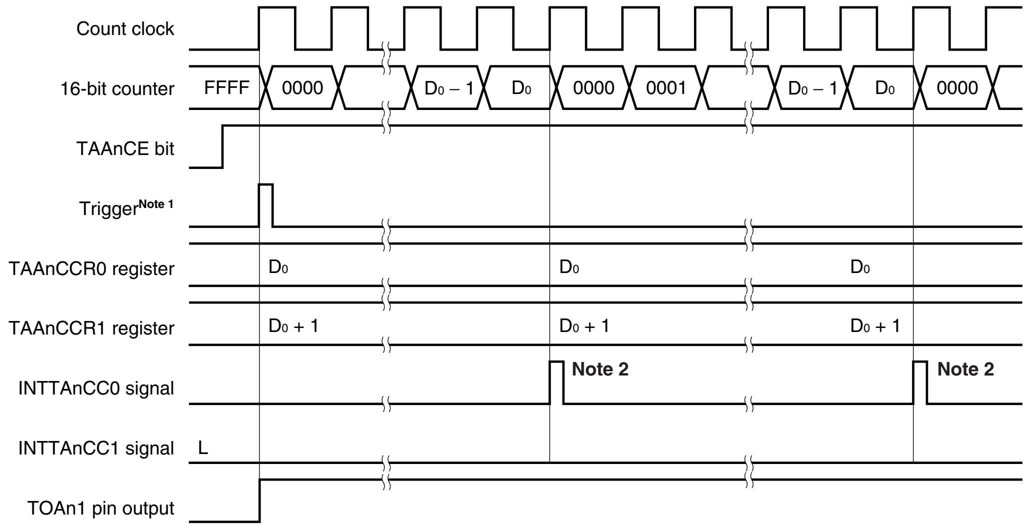


Notes 1. A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

2. The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 2

To output a 100% waveform, set a value of (set value of TAAnCCR0 register + 1) to the TAAnCCR1 register. If the set value of the TAAnCCR0 register is FFFFH, 100% output cannot be produced.



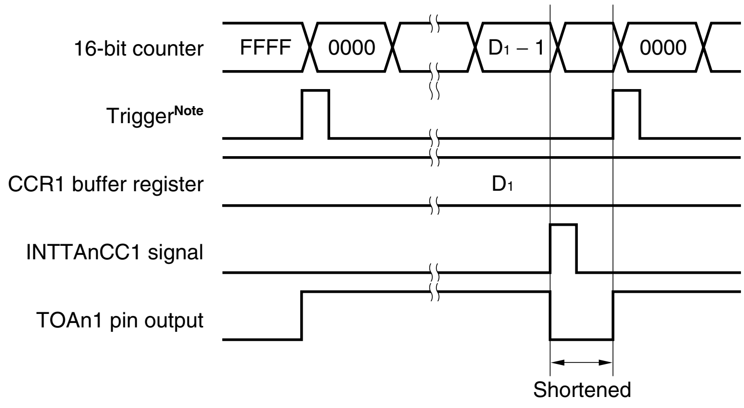
Notes 1. A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

2. The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 2

(c) Conflict between trigger detection and match with CCR1 buffer register

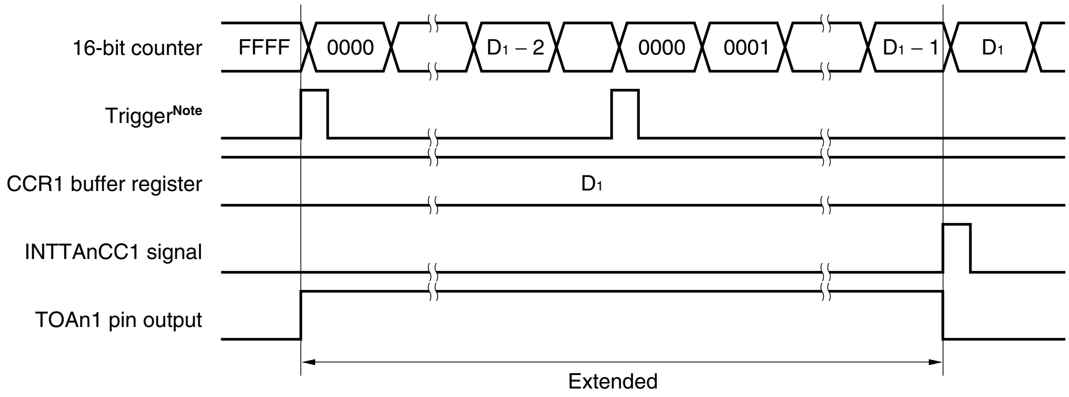
If the trigger is detected immediately after the INTTAnCC1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOAn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Note A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

Remark n = 0 to 2

If the trigger is detected immediately before the INTTAnCC1 signal is generated, the INTTAnCC1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOAn1 pin remains active. Consequently, the active period of the PWM waveform is extended.

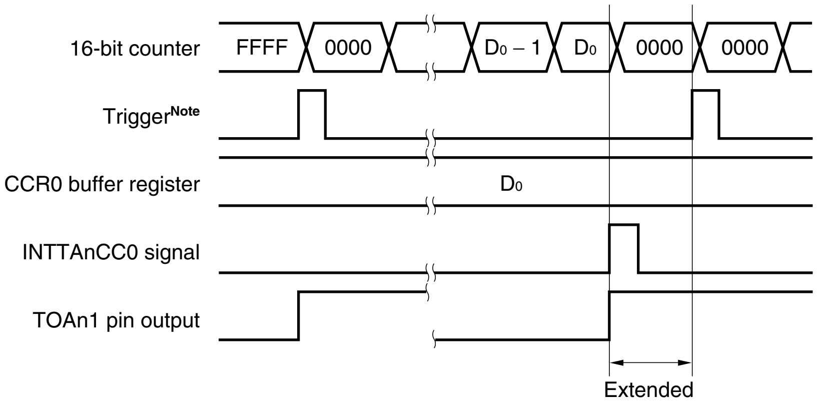


Note A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

Remark n = 0 to 2

(d) Conflict between trigger detection and match with CCR0 buffer register

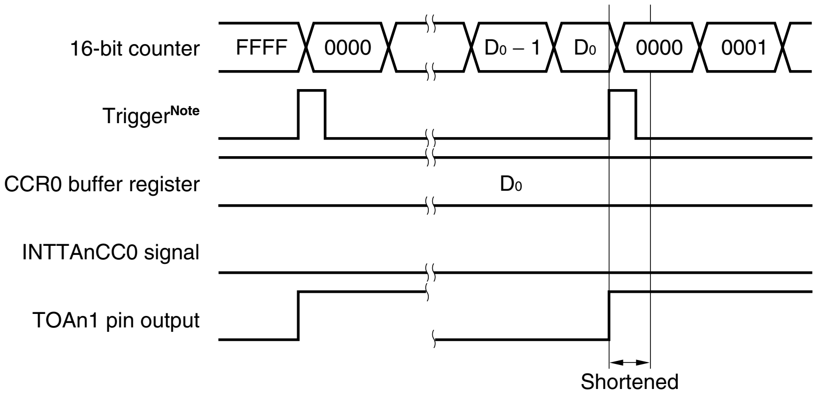
If the trigger is detected immediately after the INTTAnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOAn1 pin is extended by time from generation of the INTTAnCC0 signal to trigger detection.



Note A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

Remark n = 0 to 2

If the trigger is detected immediately before the INTTAnCC0 signal is generated, the INTTAnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOAn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

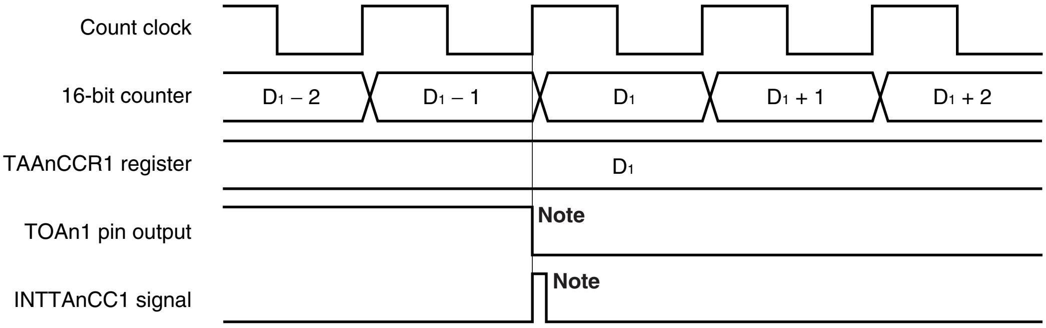


Note A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

Remark n = 0 to 2

(e) Generation timing of compare match interrupt request signal (INTTAnCC1)

The timing of generation of the INTTAnCC1 signal in the external trigger pulse output mode differs from the timing of INTTAnCC1 signals in other mode; the INTTAnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAnCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 2

Usually, the INTTAnCC1 signal is generated in synchronization with the next count-up, after the count value of the 16-bit counter matches the value of the TAAnCCR1 register.

In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOAn1 pin.

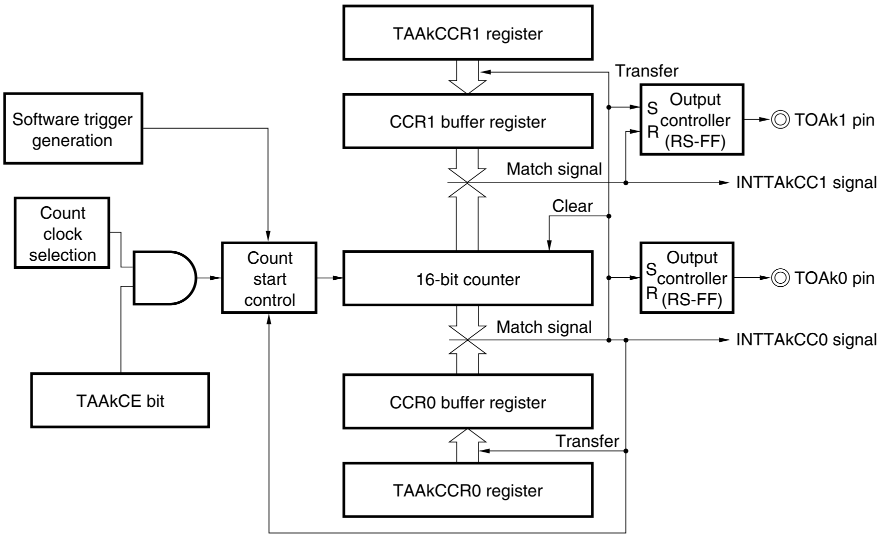
6.6.4 One-shot pulse output mode (TAAnMD2 to TAAnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter AA waits for a trigger when the TAAnCTL0.TAAnCE bit is set to 1.

For TAA0 and TAA1, the counter starts incrementing when a software trigger is detected and a one shot pulse is output from the TOAk1 pin. When the software trigger is used, the TOAk0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

For TAA2, the counter starts incrementing when the valid edge of the external trigger input (TIA20) is detected and a one shot pulse is output from the TOA21 pin. Pulses can also be output by generating a software trigger instead of using the external trigger input. When the software trigger is used, the TOA20 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

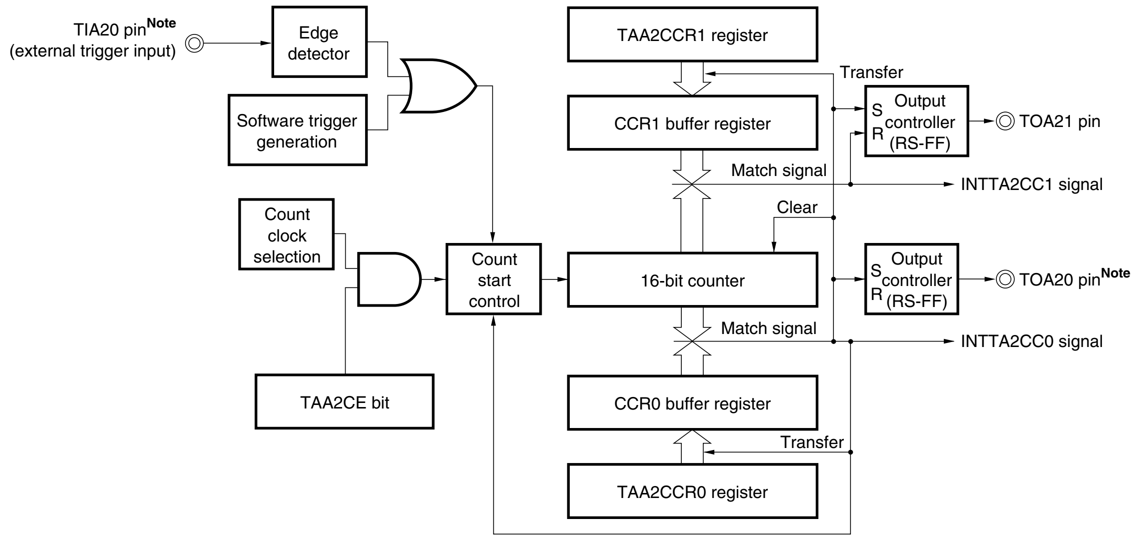
Figure 6-27. Configuration of TAA0 and TAA1 in One-Shot Pulse Output Mode



Caution In the one-shot pulse output mode, select the internal clock as the count clock (by clearing the TAAkCTL1.TAAkEEE bit to 0).

Remark k = 0, 1

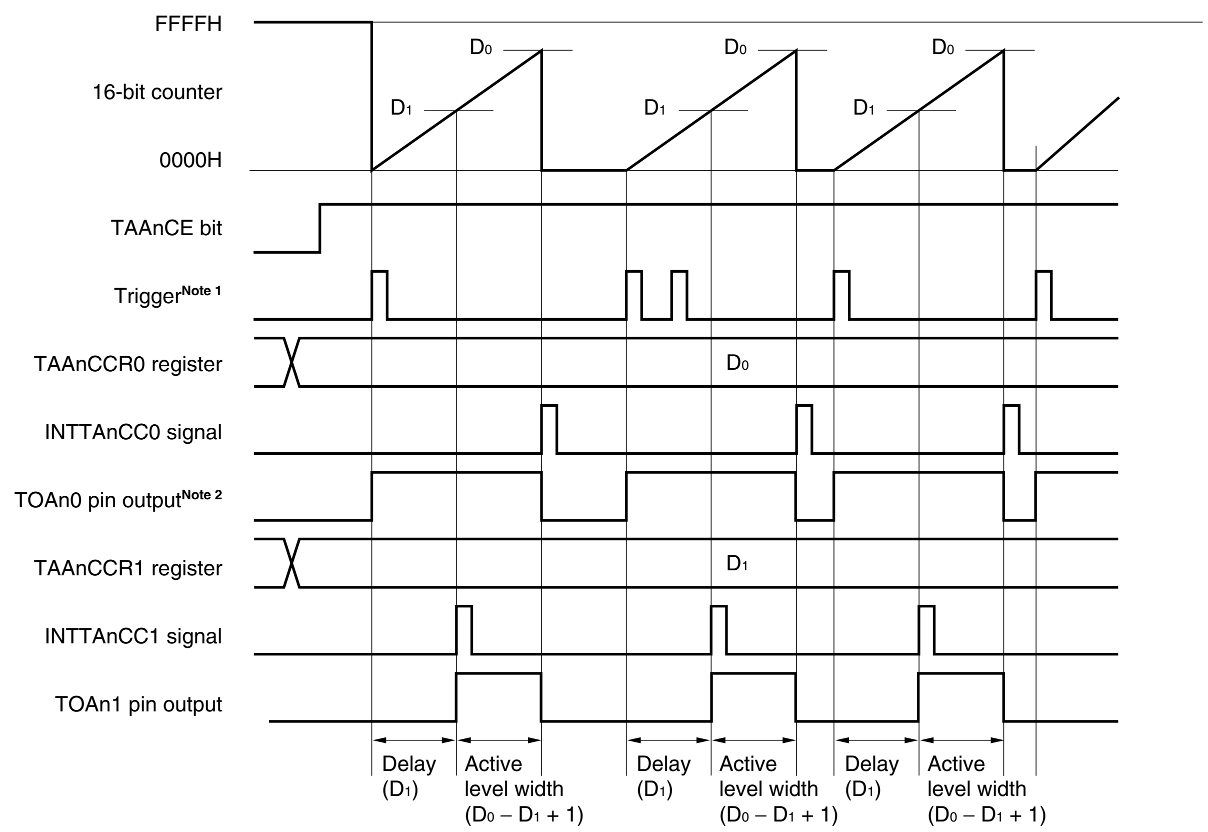
Figure 6-28. Configuration of TAA2 in One-Shot Pulse Output Mode



Note Because the external trigger input pin (TIA20) and timer output pin (TOA20) are the same pin, the two functions cannot be used at the same time.

Caution In the one-shot pulse output mode, select the internal clock as the count clock (by clearing the TAA2CTL1.TAA2EEE bit to 0).

Figure 6-29. Basic Timing in One-Shot Pulse Output Mode



Notes 1. A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

2. For TAA2, this function can only be used by using a software trigger.

When the TAAnCE bit is set to 1, 16-bit timer/event counter AA waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOAn1 pin. After the one-shot pulse is output, the 16-bit counter is cleared to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

Output delay period = (Set value of TAAnCCR1 register) × Count clock cycle

Active level width = (Set value of TAAnCCR0 register − Set value of TAAnCCR1 register + 1) × Count clock cycle

The compare match interrupt request signal (INTTAnCC0) is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal (INTTAnCC1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The valid edge of an external trigger input (TIA20 pin) or setting the software trigger (TAAnCTL1.TAAnEST bit) to 1 is used as the trigger.

Remark n = 0 to 2

Figure 6-30. Setting of Registers in One-Shot Pulse Output Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TAAnCKS2 to TAAnCKS0: Select count clock

TAAnCE:
0: Stop counting
1: Enable counting

(b) TAAn control register 1 (TAAnCTL1)

	TAAaSYE	TAAnEST	TAA2EEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0/1	0	0	0	0	1	1

TAAnMD2 to TAAnMD0:
0, 1, 1:
One-shot pulse output mode

TAA2EEE:
0: Operate on count clock selected by TAA2CKS0 to TAA2CKS2 bits

TAAnEST:
Generate software trigger when 1 is written

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

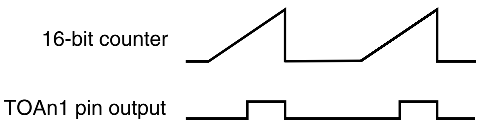
TAAnOE0:
0: Disable TOAn0 pin output
1: Enable TOAn0 pin output

TAAnOL0:
Setting of TOAn0 pin output level while waiting for external trigger
0: Low level
1: High level

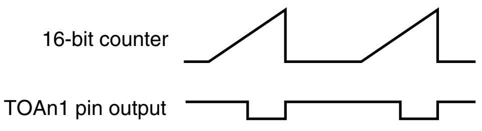
TAAnOE1:
0: Disable TOAn1 pin output
1: Enable TOAn1 pin output

TAAnOL1:
Setting of TOAn1 pin output level while waiting for external trigger
0: Low level
1: High level

- When TAAnOL1 bit = 0



- When TAAnOL1 bit = 1



Note Clear this bit to 0 when the TOAn0 pin is not used in the one-shot pulse output mode.

Figure 6-30. Setting of Registers in One-Shot Pulse Output Mode (2/2)

(d) TAA2 I/O control register 2 (TAA2IOC2)

							TAA2EES1	TAA2EES0	TAA2ETS1	TAA2ETS0
TAA2IOC2	0	0	0	0	0	0	0/1	0/1

Select valid edge of external trigger input (TIA20 pin)

(e) TAAn counter read buffer register (TAAnCNT)

The value of the 16-bit counter can be read by reading the TAAnCNT register.

(f) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

If D_0 is set to the TAAnCCR0 register and D_1 to the TAAnCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width = $(D_0 - D_1 + 1) \times$ Count clock cycle

Output delay period = $D_1 \times$ Count clock cycle

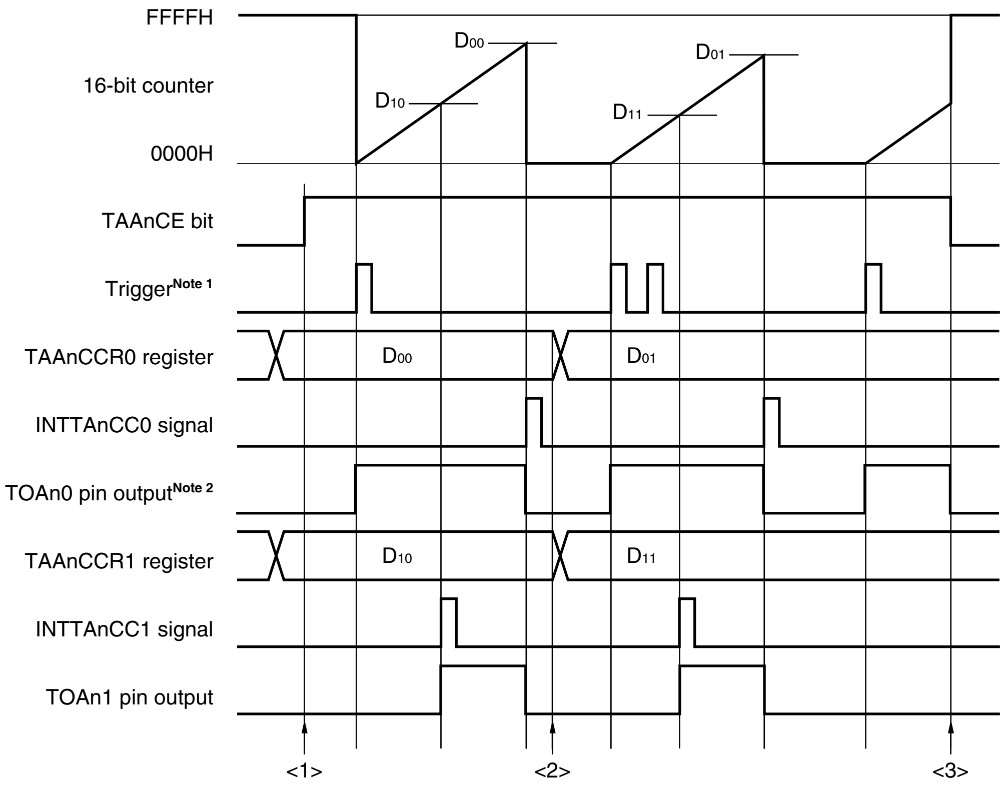
Caution One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TAAnCCR1 register is greater than that set in the TAAnCCR0 register.

Remarks 1. TAA2 I/O control register 1 (TAA2IOC1) and TAAn option register 0 (TAAnOPT0) are not used in the one-shot pulse output mode.

2. n = 0 to 2
a = 0, 1

(1) Operation flow in one-shot pulse output mode

Figure 6-31. Software Processing Flow in One-Shot Pulse Output Mode (1/2)

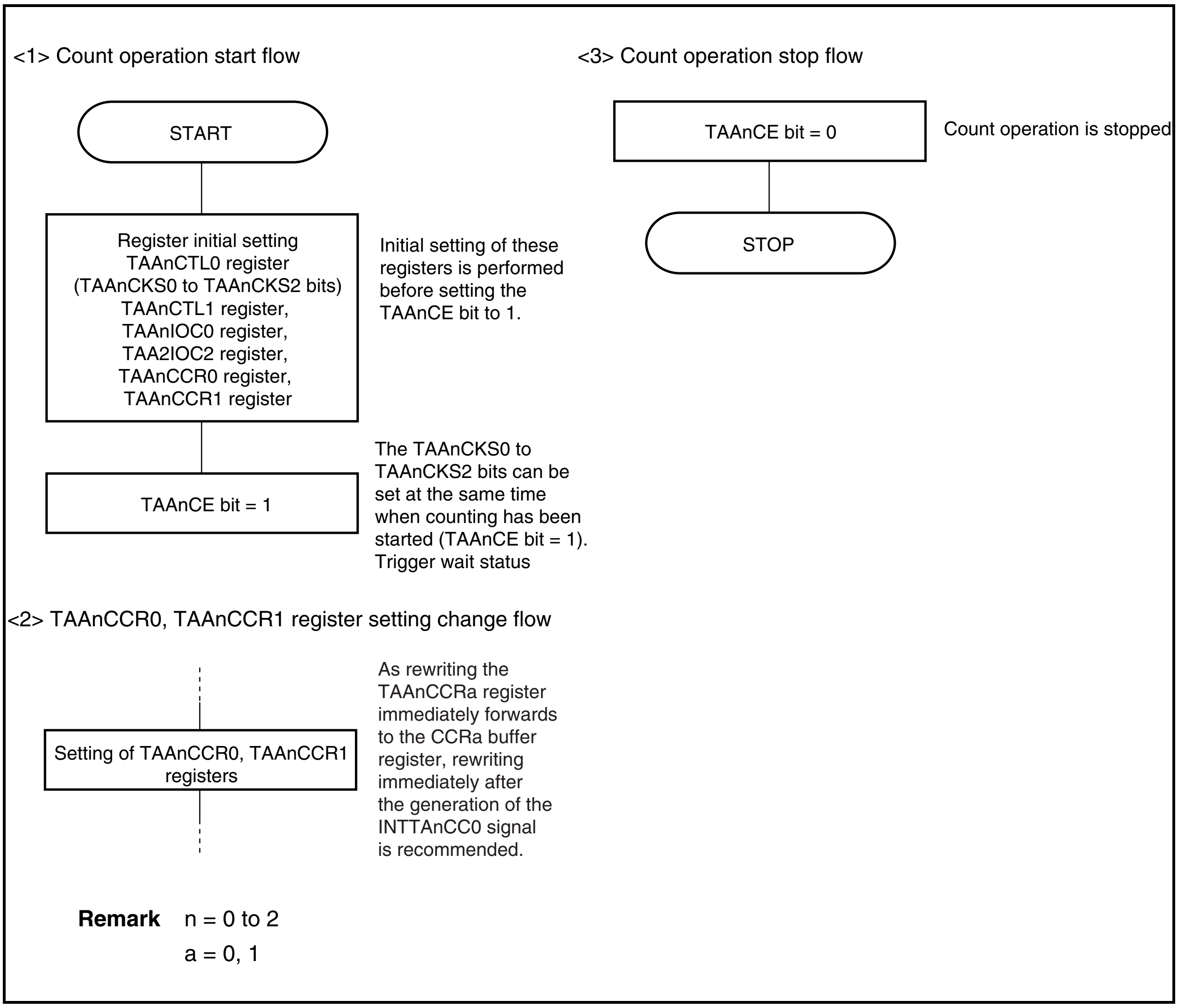


Notes 1. A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

2. For TAA2, this function can only be used by using a software trigger.

Remark n = 0 to 2

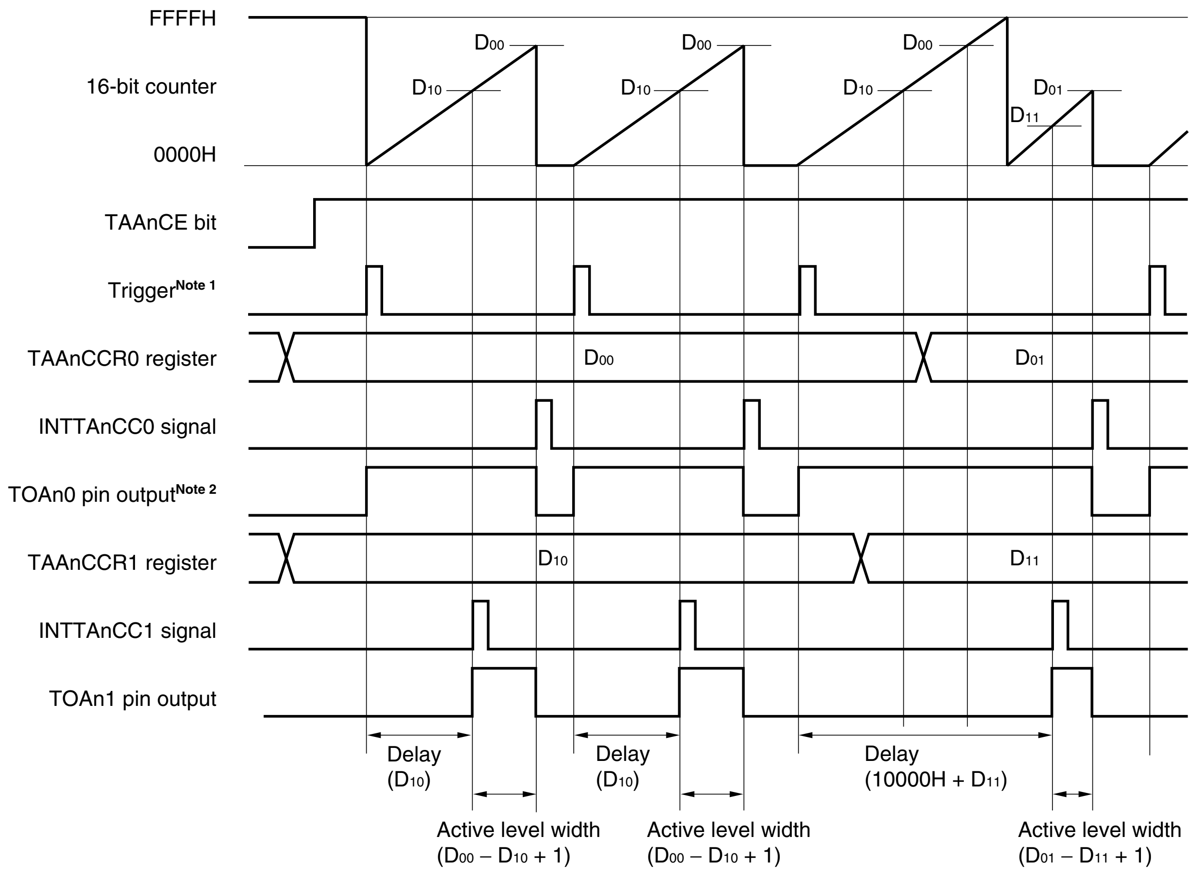
Figure 6-31. Software Processing Flow in One-Shot Pulse Output Mode (2/2)


<1> Count operation start flow
START
Register initial setting
TAAnCTL0 register
(TAAnCKS0 to TAAnCKS2 bits)
TAAnCTL1 register,
TAAnIOC0 register,
TAA2IOC2 register,
TAAnCCR0 register,
TAAnCCR1 register
Initial setting of these registers is performed before setting the TAAnCE bit to 1.
TAAnCE bit = 1
The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting has been started (TAAnCE bit = 1).
Trigger wait status
<2> TAAnCCR0, TAAnCCR1 register setting change flow
Setting of TAAnCCR0, TAAnCCR1 registers
As rewriting the TAAnCCRa register immediately forwards to the CCRa buffer register, rewriting immediately after the generation of the INTTAnCC0 signal is recommended.
<3> Count operation stop flow
TAAnCE bit = 0
Count operation is stopped
STOP
Remark n = 0 to 2
a = 0, 1

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TAAnCCRa register

If the value of the TAAnCCRa register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



Notes 1. A software trigger for TAA0 and TAA1, and an external trigger input (from the TIA20 pin) for TAA2

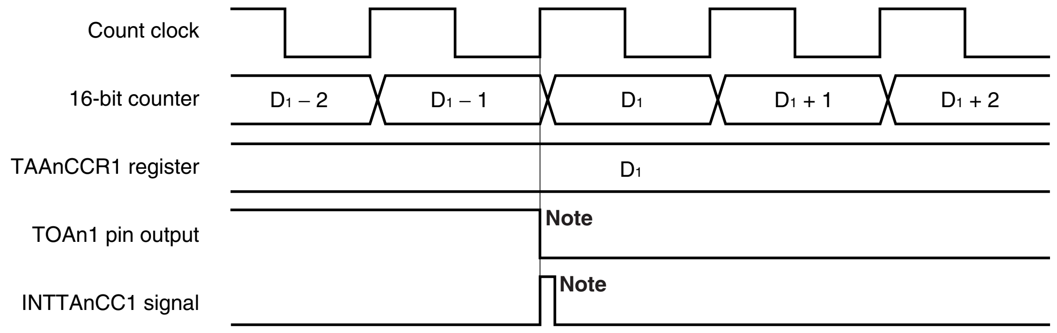
2. For TAA2, this function can only be used by using a software trigger.

When the TAAnCCR0 register is rewritten from D_{00} to D_{01} and the TAAnCCR1 register from D_{10} to D_{11} where $D_{00} > D_{01}$ and $D_{10} > D_{11}$, if the TAAnCCR1 register is rewritten when the count value of the 16-bit counter is greater than D_{11} and less than D_{10} and if the TAAnCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{11}, the counter generates the INTTAnCC1 signal and asserts the TOAn1 pin output. When the count value matches D_{01}, the counter generates the INTTAnCC0 signal, deasserts the TOAn1 pin output, and stops counting.
Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark n = 0 to 2
a = 0, 1

(b) Generation timing of compare match interrupt request signal (INTTAnCC1)

The generation timing of the INTTAnCC1 signal in the one-shot pulse output mode is different from INTTAnCC1 signals; the INTTAnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAnCCR1 register.



Note The timing is actually delayed by one operating clock (f_{XX}).

Remark n = 0 to 2

Usually, the INTTAnCC1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TAAnCCR1 register.

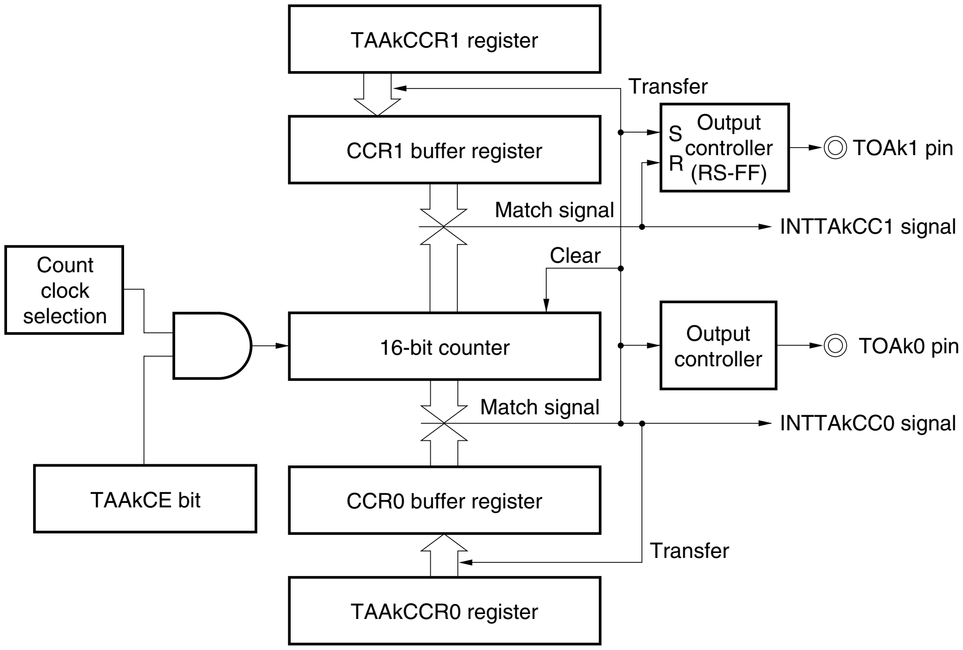
In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOAn1 pin.

6.6.5 PWM output mode (TAAnMD2 to TAAnMD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TOAn1 pin when the TAAnCTL0.TAAnCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TAAnCCR0 register + 1 as half its cycle is output from the TOAn0 pin.

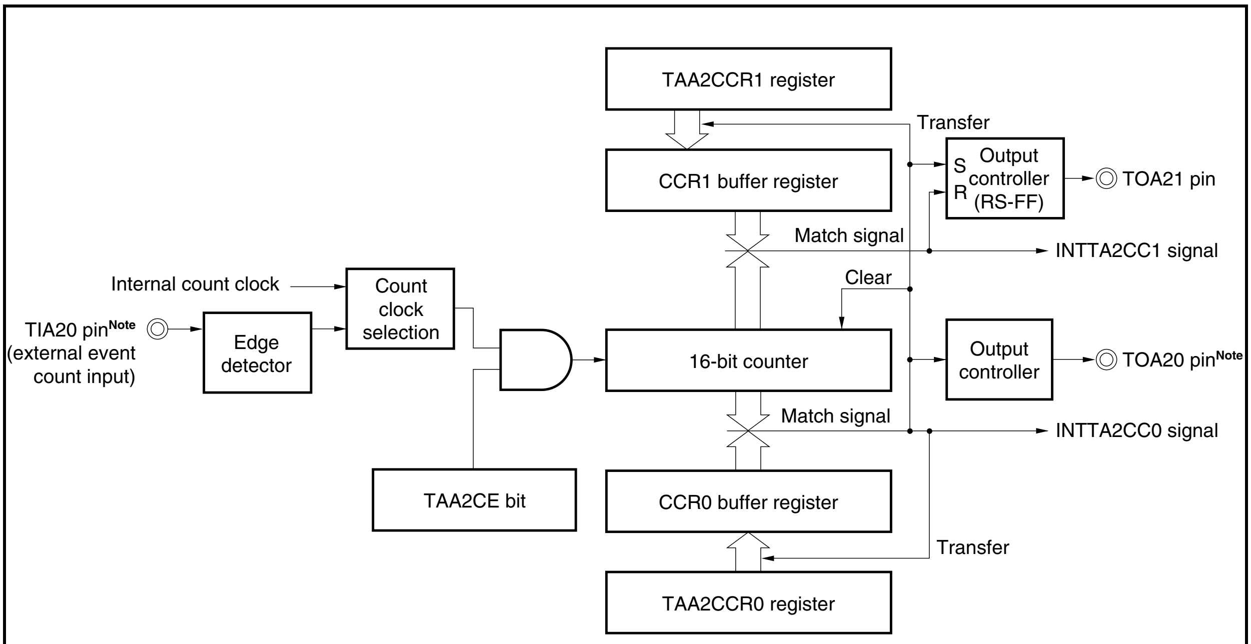
Figure 6-32. Configuration of TAA0 and TAA1 in PWM Output Mode



Caution **When TAA0 and TAA1 are in the PWM output mode, specify the internal clock as the count clock (by clearing the TAAkCTL1.TAAkEEE bit to 0).**

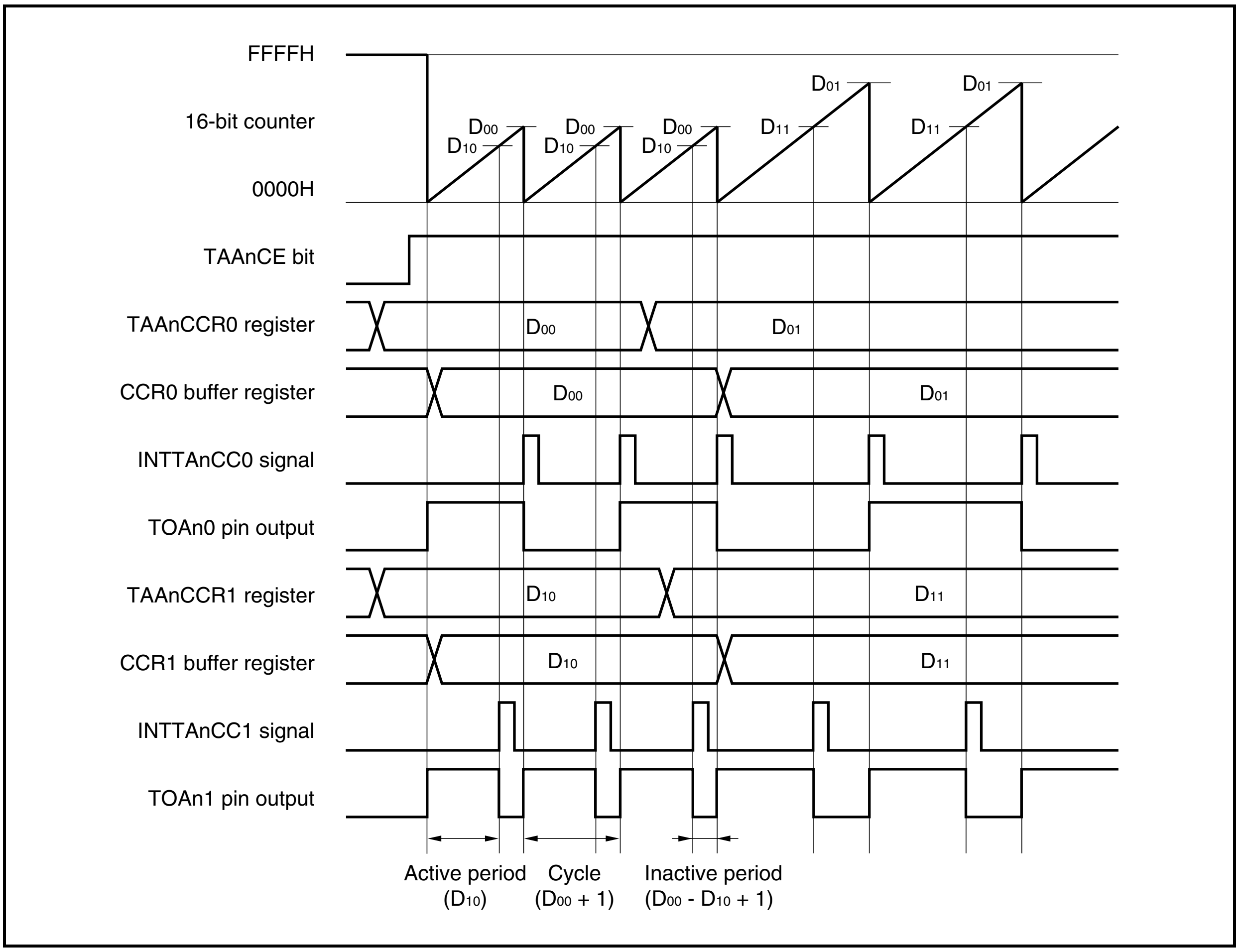
Remark k = 0, 1

Figure 6-33. Configuration of TAA2 in PWM Output Mode



Note Because the external event count input pin (TIA20) and timer output pin (TOA20) are the same pin, the two functions cannot be used at the same time.

Figure 6-34. Basic Timing in PWM Output Mode



When the TAAnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOAn1 pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TAAnCCR1 register) × Count clock cycle
Cycle = (Set value of TAAnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TAAnCCR1 register)/(Set value of TAAnCCR0 register + 1)

The PWM waveform can be changed by rewriting the TAAnCCRa register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal INTTAnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTAnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TAAnCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

Remark n = 0 to 2
a = 0, 1

Figure 6-35. Setting of Registers in PWM Output Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TAAnCKS2 to TAAnCKS0: Select count clock[Note]

TAAnCE:
0: Stop counting
1: Enable counting

Note The setting is invalid when the TAAnCTL1.TAA2EEE bit = 1.

(b) TAAn control register 1 (TAAnCTL1)

	TAAaSYE	TAAnEST	TAA2EEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0	0/1	0	0	1	0	0

TAAnMD2 to TAAnMD0:
1, 0, 0:
PWM output mode

TAA2EEE:
0: Operate on count clock selected by TAA2CKS0 to TAA2CKS2 bits
1: Count with external event count input signal

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

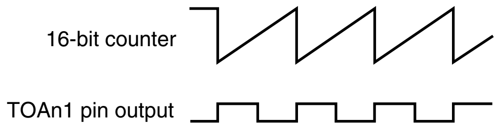
TAAnOE0:
0: Disable TOAn0 pin output
1: Enable TOAn0 pin output

TAAnOL0:
Setting of TOAn0 pin output level before count operation
0: Low level
1: High level

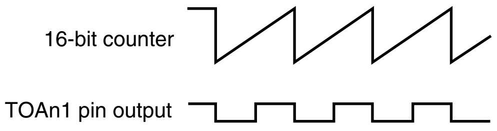
TAAnOE1:
0: Disable TOAn1 pin output
1: Enable TOAn1 pin output

TAAnOL1:
Setting of TOAn1 pin output level before count operation
0: Low level
1: High level

- When TAAnOL1 bit = 0



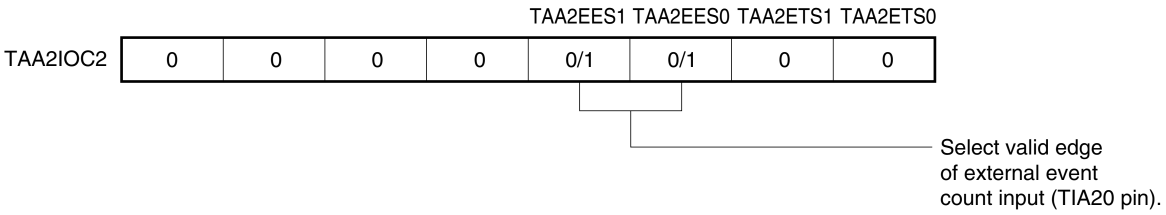
- When TAAnOL1 bit = 1



Note Clear this bit to 0 when the TOAn0 pin is not used in the PWM output mode.

Figure 6-35. Register Setting in PWM Output Mode (2/2)

(d) **TAA2 I/O control register 2 (TAA2IOC2)**



					TAA2EES1	TAA2EES0	TAA2ETS1	TAA2ETS0
TAA2IOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge of external event count input (TIA20 pin).

(e) **TAAn counter read buffer register (TAAnCNT)**

The value of the 16-bit counter can be read by reading the TAAnCNT register.

(f) **TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)**

If D_0 is set to the TAAnCCR0 register and D_1 to the TAAnCCR1 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle

Active level width = $D_1 \times$ Count clock cycle

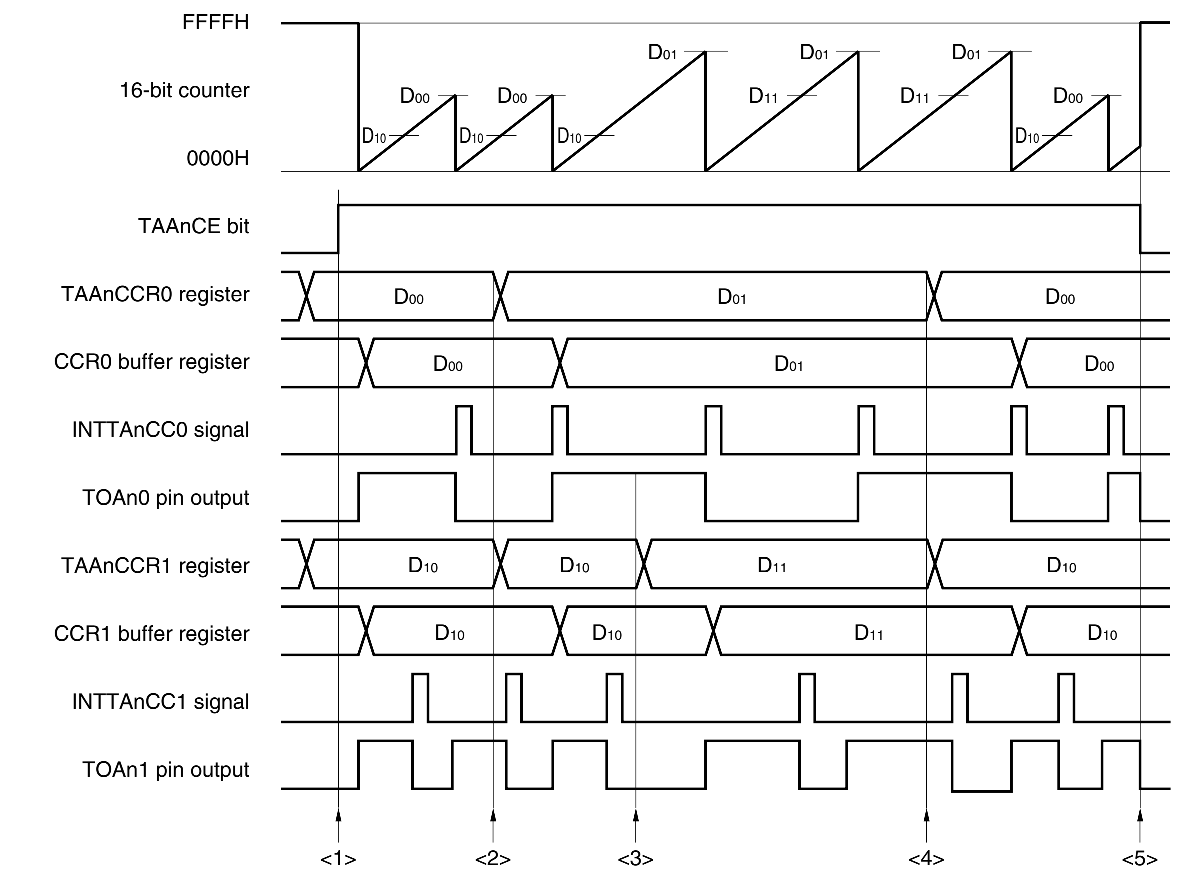
Remarks 1. TAA2 I/O control register 1 (TAA2IOC1) and TAAn option register 0 (TAAnOPT0) are not used in the PWM output mode.

2. n = 0 to 2

a = 0, 1

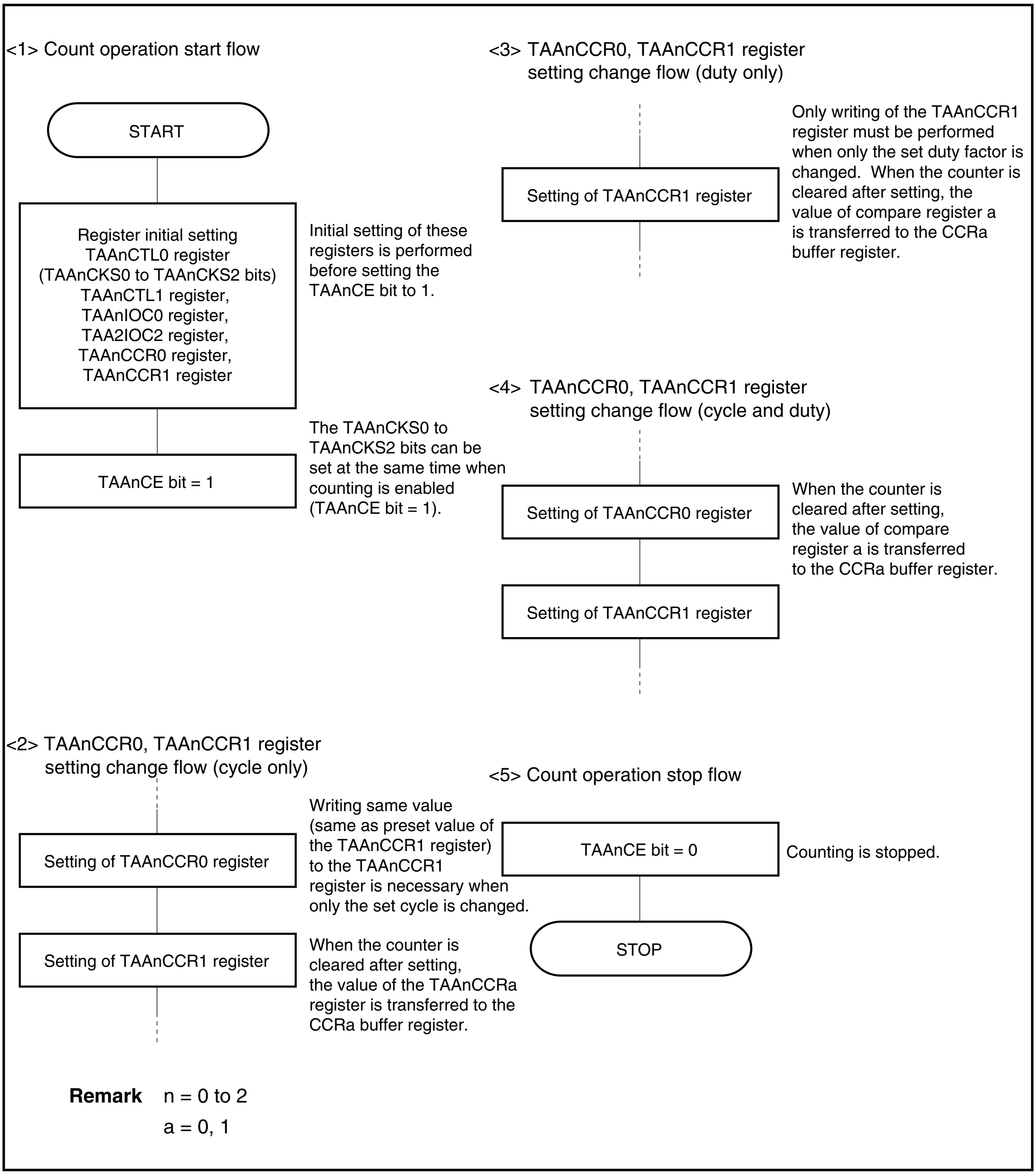
(1) Operation flow in PWM output mode

Figure 6-36. Software Processing Flow in PWM Output Mode (1/2)



Remark n = 0 to 2

Figure 6-36. Software Processing Flow in PWM Output Mode (2/2)


<1> Count operation start flow
START
Register initial setting
TAAnCTL0 register
(TAAnCKS0 to TAAnCKS2 bits)
TAAnCTL1 register,
TAAnIOC0 register,
TAA2IOC2 register,
TAAnCCR0 register,
TAAnCCR1 register
TAAnCE bit = 1
<2> TAAnCCR0, TAAnCCR1 register setting change flow (cycle only)
Setting of TAAnCCR0 register
Setting of TAAnCCR1 register
<3> TAAnCCR0, TAAnCCR1 register setting change flow (duty only)
Setting of TAAnCCR1 register
<4> TAAnCCR0, TAAnCCR1 register setting change flow (cycle and duty)
Setting of TAAnCCR0 register
Setting of TAAnCCR1 register
<5> Count operation stop flow
TAAnCE bit = 0
STOP

Initial setting of these registers is performed before setting the TAAnCE bit to 1.

The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting is enabled (TAAnCE bit = 1).

Writing same value (same as preset value of the TAAnCCR1 register) to the TAAnCCR1 register is necessary when only the set cycle is changed.

When the counter is cleared after setting, the value of the TAAnCCRa register is transferred to the CCRa buffer register.

Only writing of the TAAnCCR1 register must be performed when only the set duty factor is changed. When the counter is cleared after setting, the value of compare register a is transferred to the CCRa buffer register.

When the counter is cleared after setting, the value of compare register a is transferred to the CCRa buffer register.

Counting is stopped.

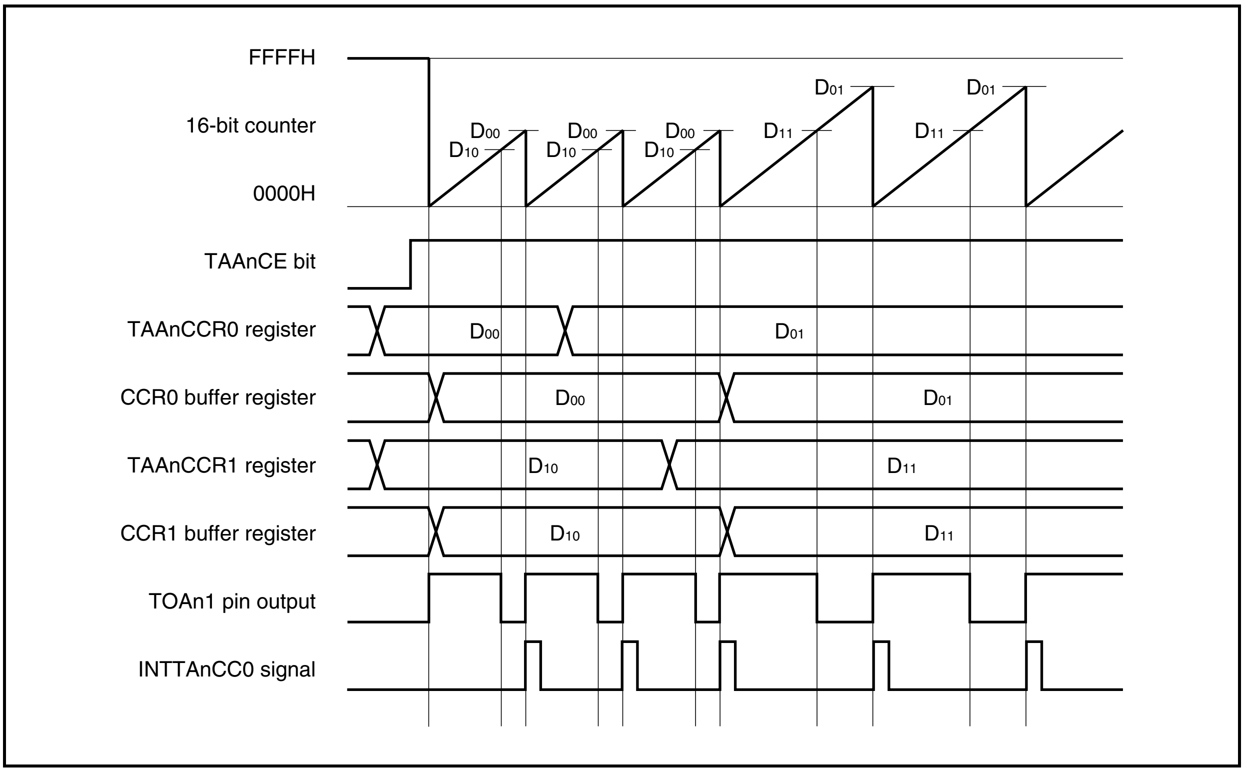
Remark n = 0 to 2
a = 0, 1

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TAAnCCR1 register last.

Rewrite the TAAnCCRa register after writing the TAAnCCR1 register after the INTTAnCC0 signal is detected.



To transfer data from the TAAnCCRa register to the CCRa buffer register, the TAAnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TAAnCCR0 register and then set the active level width to the TAAnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TAAnCCR0 register, and then write the same value (same as preset value of the TAAnCCR1 register) to the TAAnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TAAnCCR1 register has to be set.

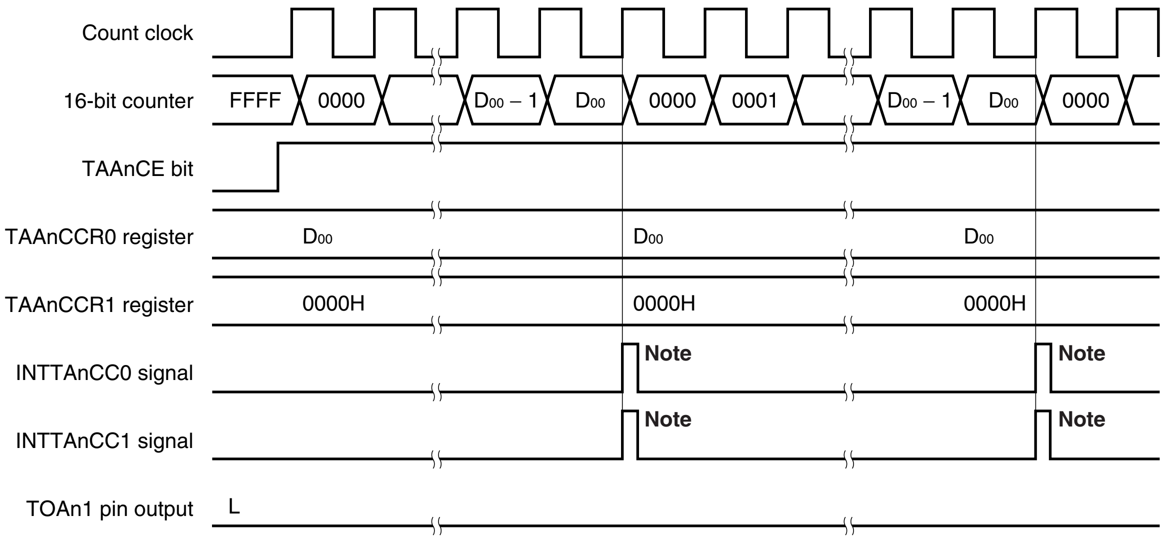
After data is written to the TAAnCCR1 register, the value written to the TAAnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TAAnCCR0 or TAAnCCR1 register again after writing the TAAnCCR1 register once, do so after the INTTAnCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TAAnCCRa register to the CCRa buffer register conflicts with writing the TAAnCCRa register.

Remark n = 0 to 2, a = 0, 1

(b) 0%/100% output of PWM waveform

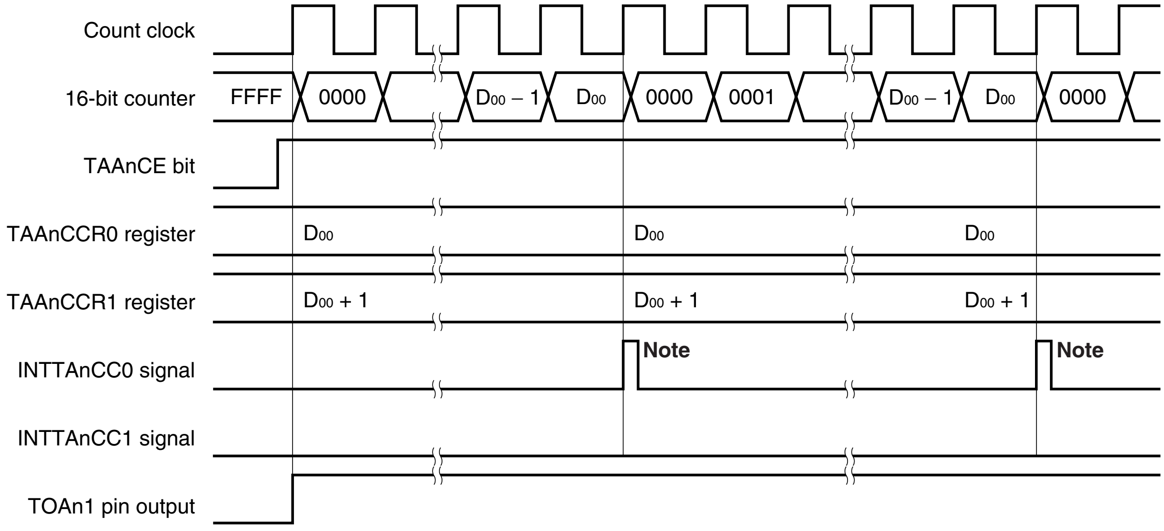
To output a 0% waveform, set the TAAnCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTAnCC0 and INTTAnCC1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 2

To output a 100% waveform, set a value of (set value of TAAnCCR0 register + 1) to the TAAnCCR1 register. If the set value of the TAAnCCR0 register is FFFFH, 100% output cannot be produced.

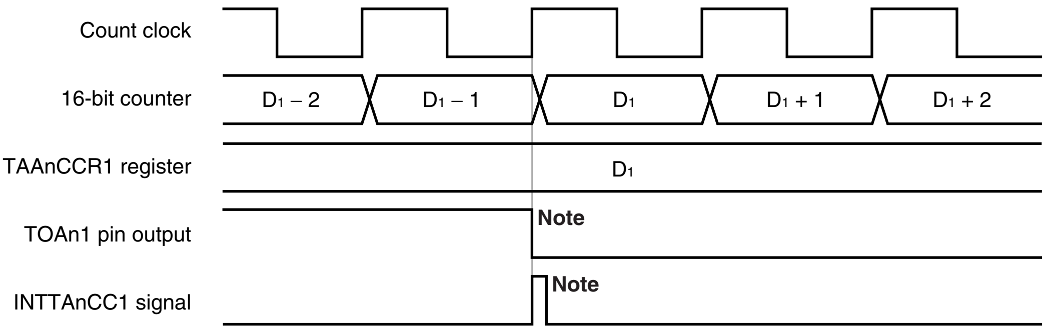


Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 2

(c) Generation timing of compare match interrupt request signal (INTTAnCC1)

The timing of generation of the INTTAnCC1 signal in the PWM output mode differs from the timing of INTTAnCC1 signals; the INTTAnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAnCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 2

Usually, the INTTAnCC1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TAAnCCR1 register.

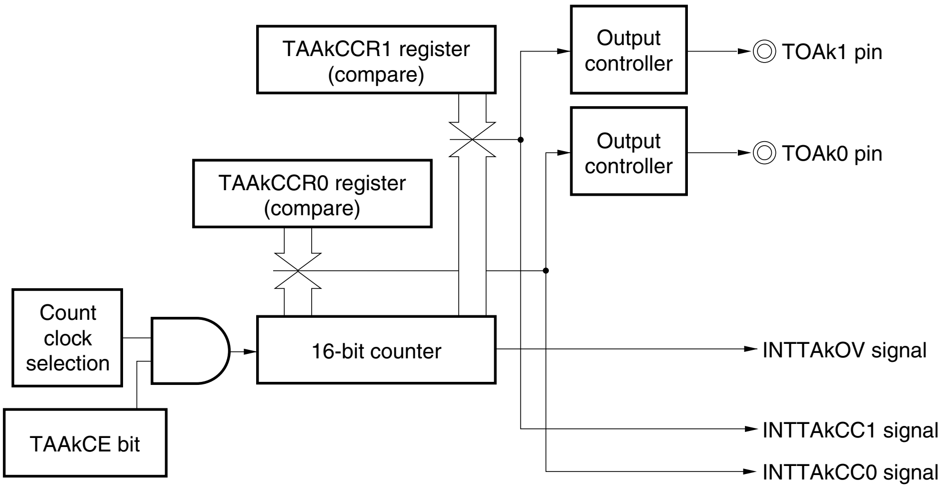
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOAn1 pin.

6.6.6 Free-running timer mode (TAAnMD2 to TAAnMD0 bits = 101)

The compare function is valid for all of TAA0 to TAA2. The capture function is valid only for TAA2.

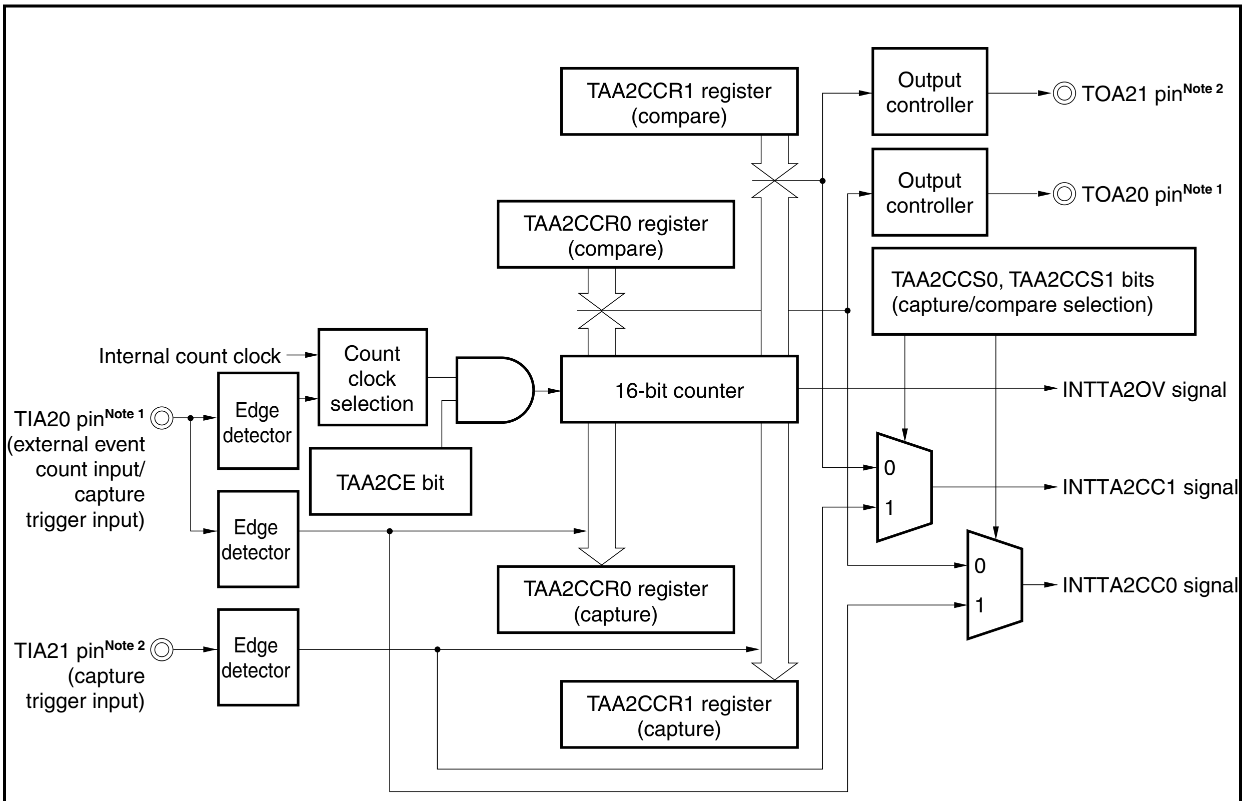
In the free-running timer mode, 16-bit timer/event counter AA starts counting when the TAAnCTL0.TAAnCE bit is set to 1. At this time, the TAA2CCR0 and TAA2CCR1 registers can be used as compare registers or capture registers, depending on the setting of the TAA2OPT0.TAA2CCS0 and TAA2OPT0.TAA2CCS1 bits.

Figure 6-37. Configuration of TAA0 and TAA1 in Free-Running Timer Mode



Remark k = 0, 1

Figure 6-38. Configuration of TAA2 in Free-Running Timer Mode


TAA2CCR1 register (compare)
Output controller
TOA21 pinNote 2
Output controller
TOA20 pinNote 1
TAA2CCR0 register (compare)
TAA2CCS0, TAA2CCS1 bits (capture/compare selection)
Internal count clock
Count clock selection
16-bit counter
INTTA2OV signal
TIA20 pinNote 1 (external event count input/ capture trigger input)
Edge detector
TAA2CE bit
0
1
INTTA2CC1 signal
Edge detector
0
1
INTTA2CC0 signal
TAA2CCR0 register (capture)
TIA21 pinNote 2 (capture trigger input)
Edge detector
TAA2CCR1 register (capture)

Notes 1. Because the external event count input pin (TIA20), capture trigger input pin (TIA20), and timer output pin (TOA20) are the same pin, the two or more functions cannot be used at the same time.

2. Because the capture trigger input pin (TIA21) and timer output pin (TOA21) are the same pin, the two functions cannot be used at the same time.

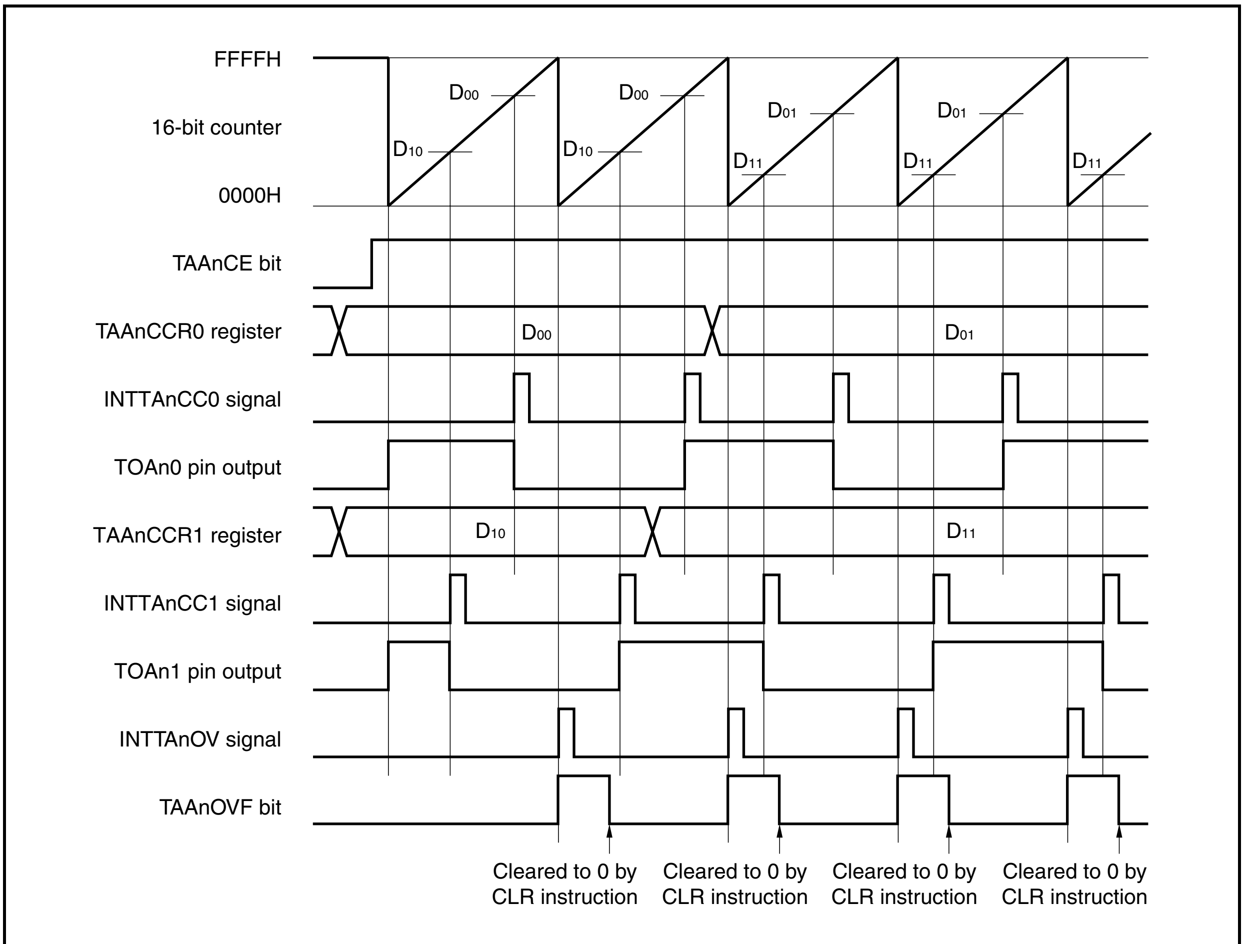
- Compare operation

  When the TAAnCE bit is set to 1, 16-bit timer/event counter AA starts counting, and the output signal of the TOAna pin is inverted. When the count value of the 16-bit counter later matches the set value of the TAAnCCRa register, a compare match interrupt request signal (INTTAnCCa) is generated, and the output signal of the TOAna pin is inverted.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTAnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TAAnOPT0.TAAnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

  The TAAnCCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time by anytime write, and compared with the count value.

Figure 6-39. Basic Timing in Free-Running Timer Mode (Compare Function)



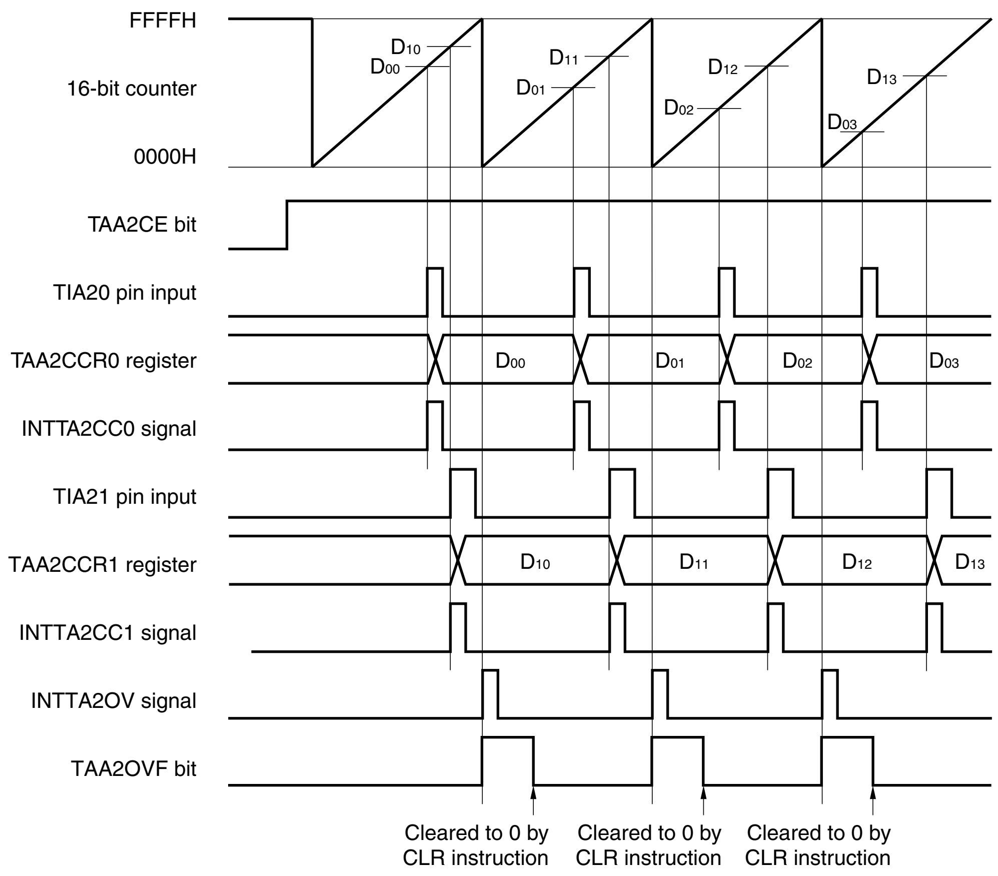
Remark n = 0 to 2
a = 0, 1

- Capture operation

  When the TAA2CE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIA2a pin is detected, the count value of the 16-bit counter is stored in the TAA2CCRa register, and a capture interrupt request signal (INTTA2CCa) is generated.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTA2OV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TAA2OPT0.TAA2OVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

Figure 6-40. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark a = 0, 1

Figure 6-41. Register Setting in Free-Running Timer Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TAAnCKS2 to TAAnCKS0: Select count clock[Note]

TAAnCE:
0: Stop counting
1: Enable counting

Note The setting is invalid when the TAA2CTL1.TAA2EEE bit = 1

(b) TAAn control register 1 (TAAnCTL1)

	TAAaSYE	TAAnEST	TAA2EEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0	0/1	0	0	1	0	1

TAAnMD2 to TAAnMD0:
1, 0, 1:
Free-running timer mode

TAA2EEE:
0: Operate with count clock selected by TAA2CKS0 to TAA2CKS2 bits
1: Count on external event count input signal

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0/1	0/1	0/1	0/1

TAAnOE0:
0: Disable TOAn0 pin output
1: Enable TOAn0 pin output

TAAnOL0:
Setting of TOAn0 pin output level before count operation
0: Low level
1: High level

TAAnOE1:
0: Disable TOAn1 pin output
1: Enable TOAn1 pin output

TAAnOL1:
Setting of TOAn1 pin output level before count operation
0: Low level
1: High level

(d) TAA2 I/O control register 1 (TAA2IOC1)

					TAA2IS3	TAA2IS2	TAA2IS1	TAA2IS0
TAA2IOC1	0	0	0	0	0/1	0/1	0/1	0/1

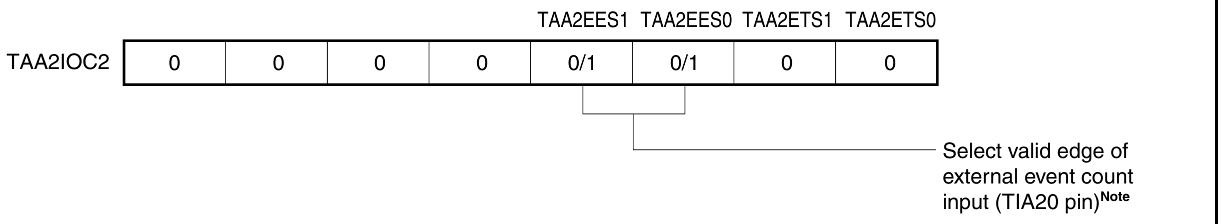
TAA2IS1, TAA2IS0: Select valid edge of TIA20 pin input[Note]

TAA2IS3, TAA2IS2: Select valid edge of TIA21 pin input[Note]

Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

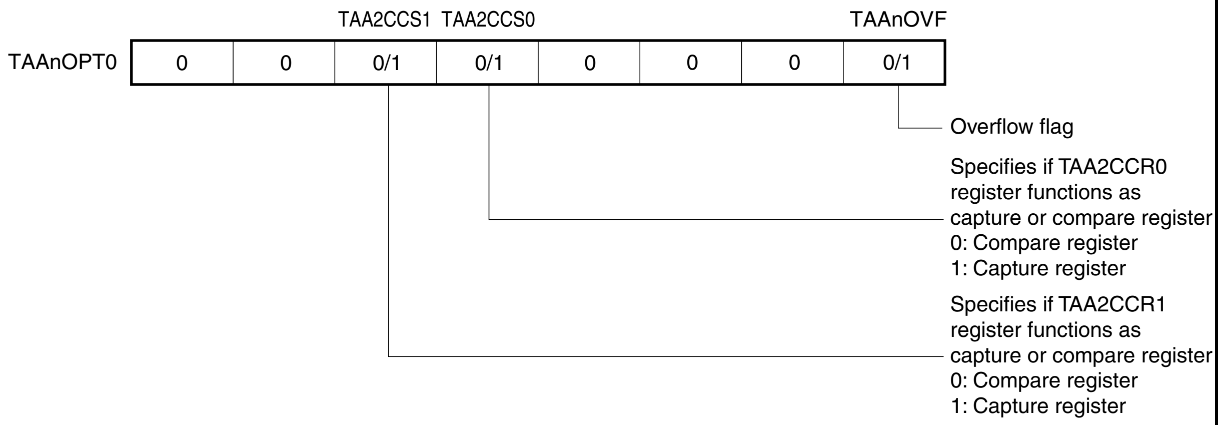
Figure 6-41. Register Setting in Free-Running Timer Mode (2/2)

(e) TAA2 I/O control register 2 (TAA2IOC2)


TAA2EES1
TAA2EES0
TAA2ETS1
TAA2ETS0
TAA2IOC2
0
0
0
0
0/1
0/1
0
0
Select valid edge of external event count input (TIA20 pin)Note

Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(f) TAAn option register 0 (TAAnOPT0)


TAA2CCS1
TAA2CCS0
TAAnOVF
TAAnOPT0
0
0
0/1
0/1
0
0
0
0/1
Overflow flag
Specifies if TAA2CCR0 register functions as capture or compare register
0: Compare register
1: Capture register
Specifies if TAA2CCR1 register functions as capture or compare register
0: Compare register
1: Capture register

(g) TAAn counter read buffer register (TAAnCNT)

The value of the 16-bit counter can be read by reading the TAAnCNT register.

(h) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

These registers function as capture registers or compare registers depending on the setting of the TAA2OPT0.TAA2CCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIA2a pin is detected.

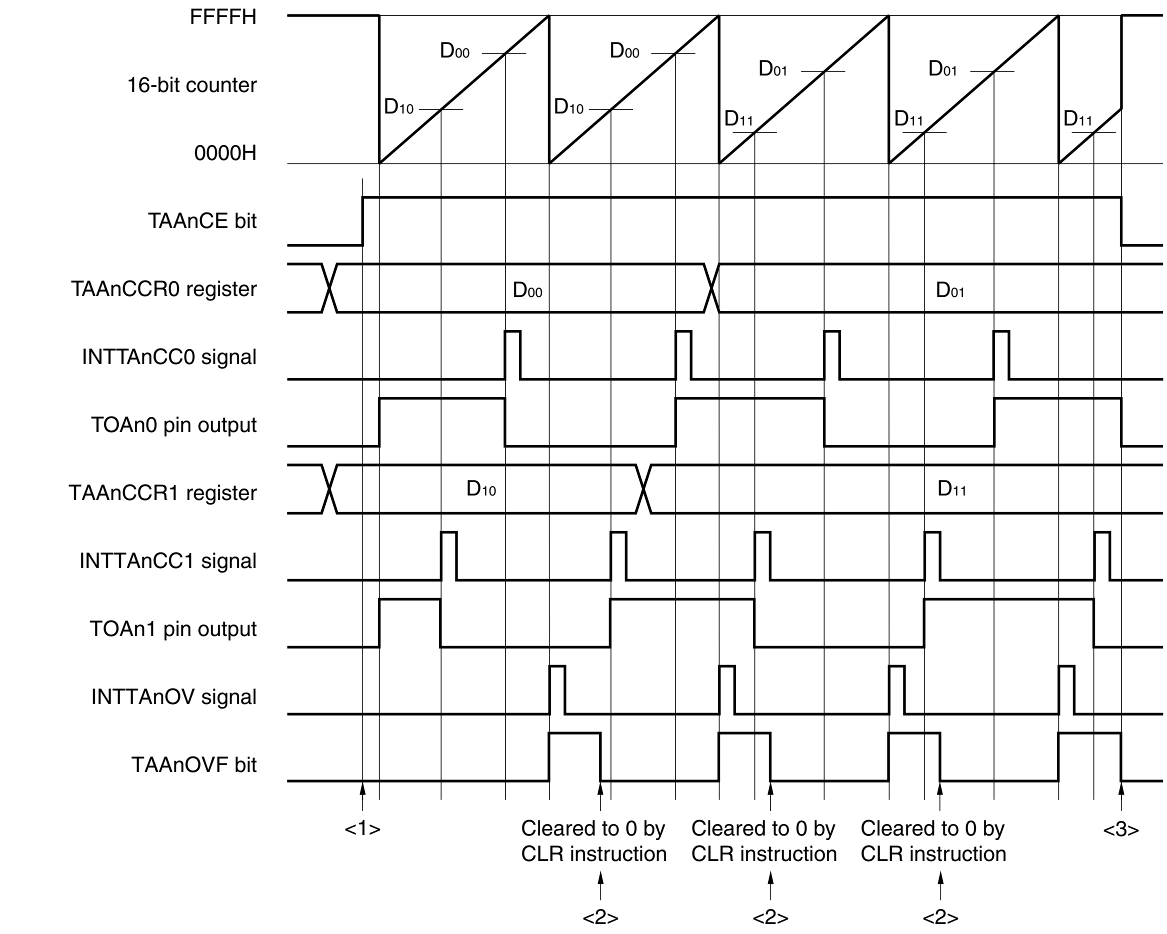
When the registers function as compare registers and when D_a is set to the TAAnCCRa register, the INTTAnCCa signal is generated when the counter reaches (D_a + 1), and the output signals of the TOAn0 and TOAn1 pins are inverted.

Remark n = 0 to 2
a = 0, 1

(1) Operation flow in free-running timer mode

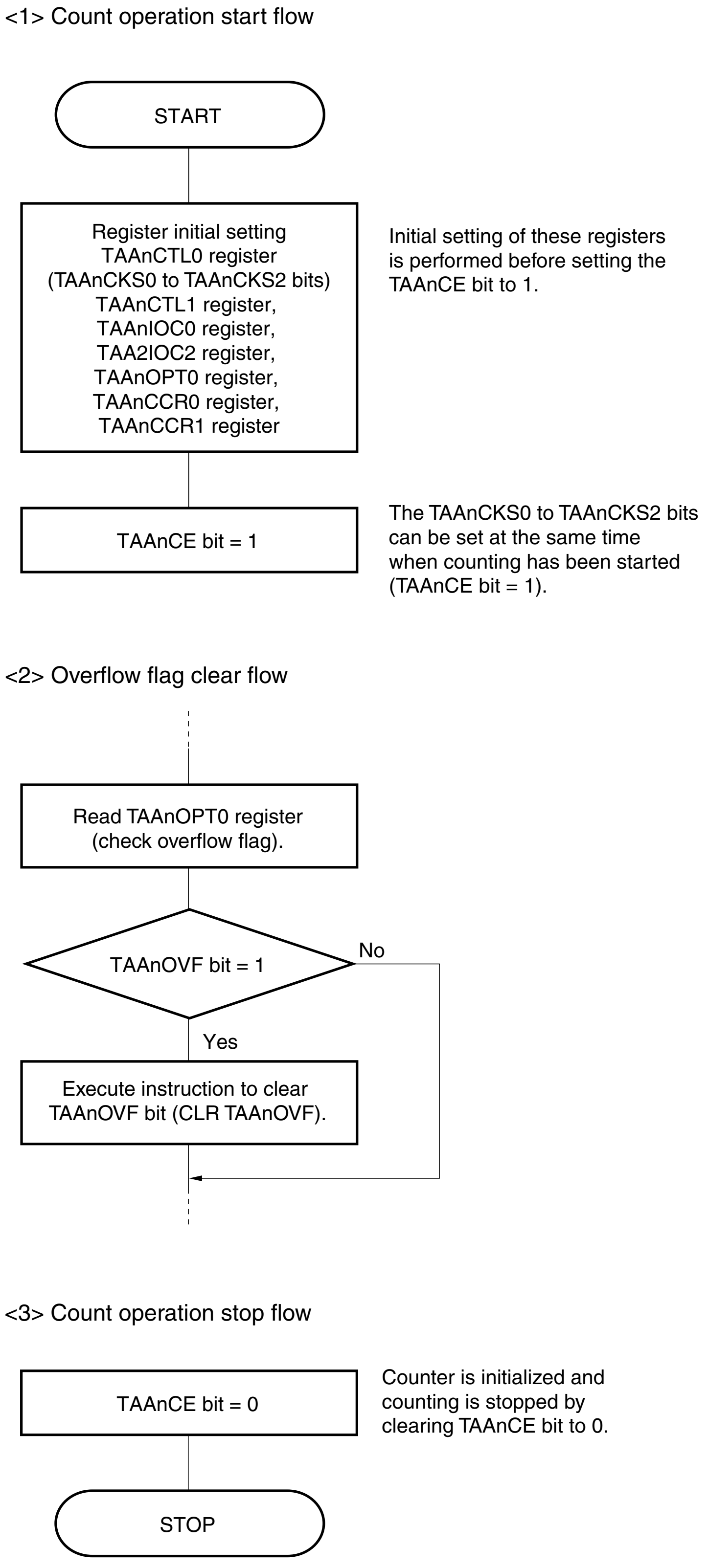
(a) When using capture/compare register as compare register

Figure 6-42. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



Remark n = 0 to 2

Figure 6-42. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)


<1> Count operation start flow
START
Register initial setting
TAAnCTL0 register
(TAAnCKS0 to TAAnCKS2 bits)
TAAnCTL1 register,
TAAnIOC0 register,
TAA2IOC2 register,
TAAnOPT0 register,
TAAnCCR0 register,
TAAnCCR1 register
Initial setting of these registers is performed before setting the TAAnCE bit to 1.
TAAnCE bit = 1
The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting has been started (TAAnCE bit = 1).
<2> Overflow flag clear flow
Read TAAnOPT0 register (check overflow flag).
TAAnOVF bit = 1
No
Yes
Execute instruction to clear TAAnOVF bit (CLR TAAnOVF).
<3> Count operation stop flow
TAAnCE bit = 0
Counter is initialized and counting is stopped by clearing TAAnCE bit to 0.
STOP

Remark n = 0 to 2

(b) When using capture/compare register as capture register

Figure 6-43. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

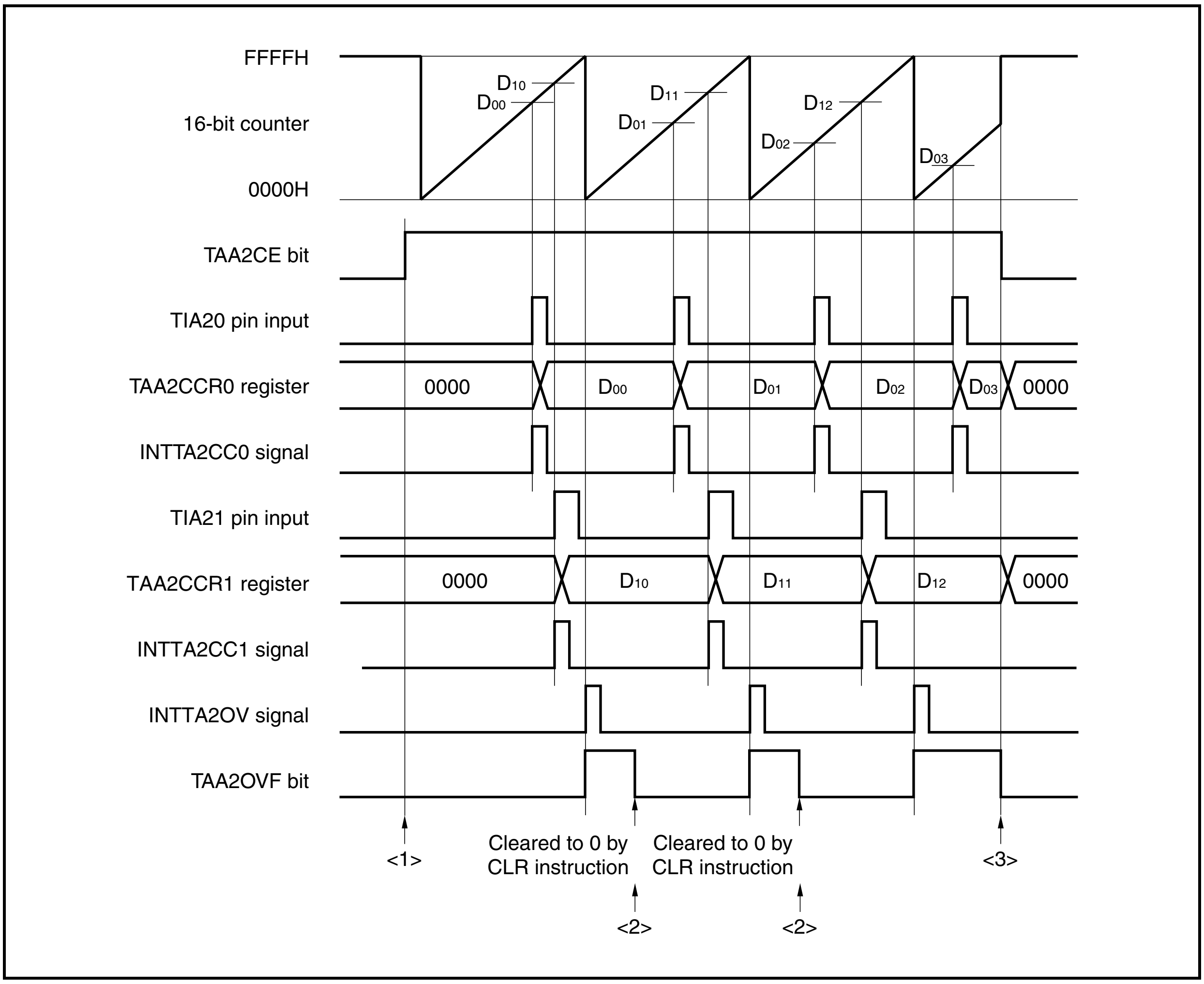
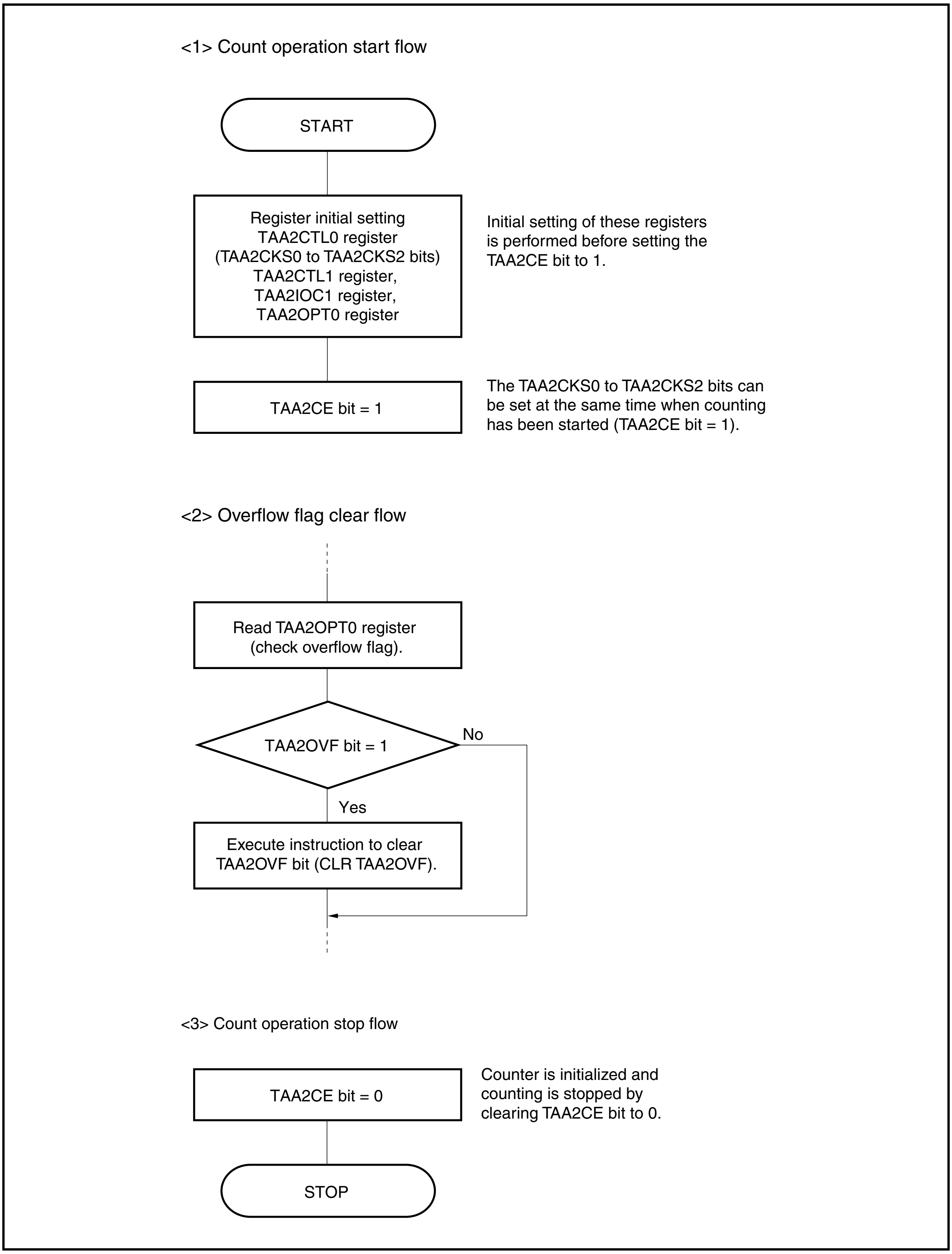


Figure 6-43. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

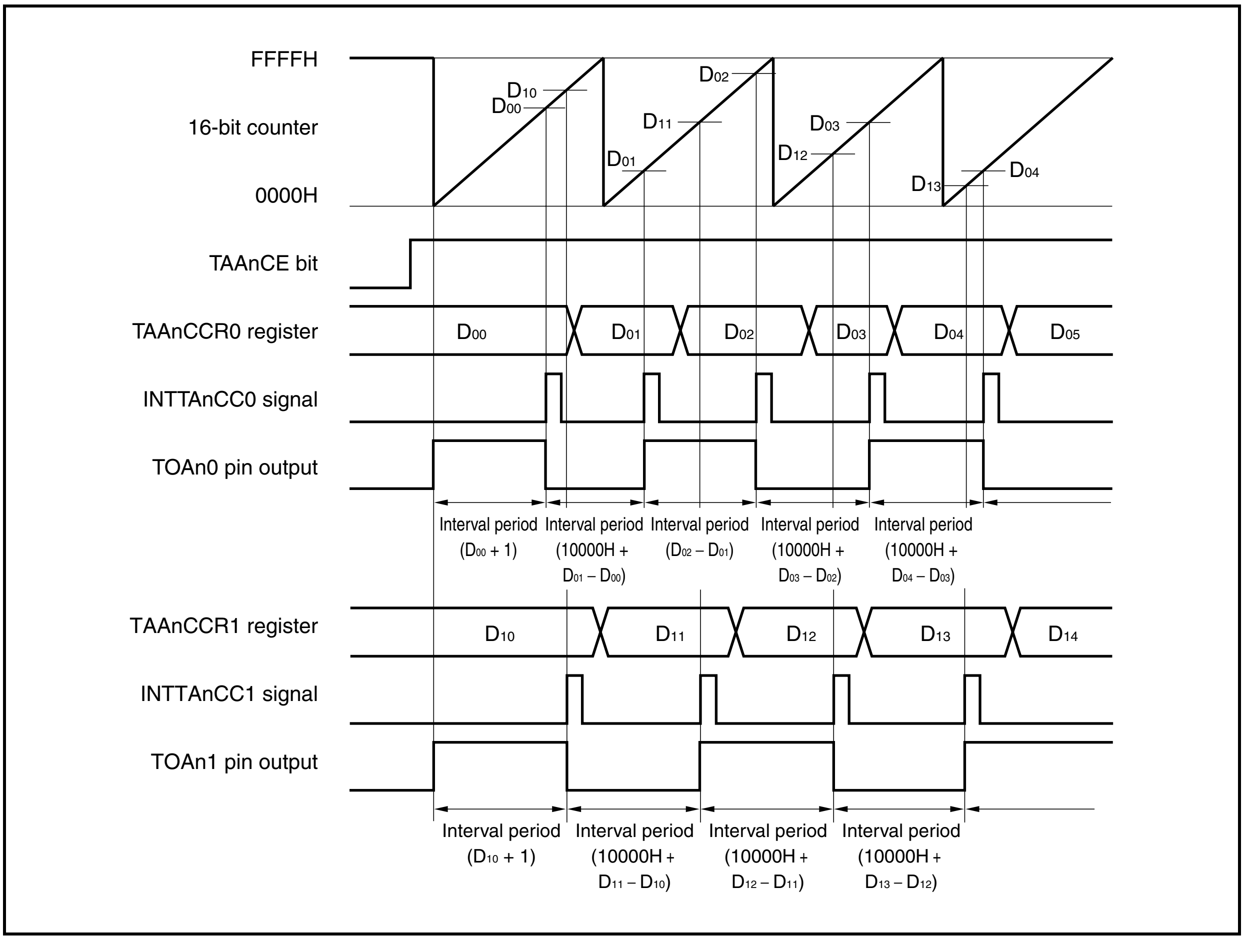


<1> Count operation start flow
START
Register initial setting
TAA2CTL0 register
(TAA2CKS0 to TAA2CKS2 bits)
TAA2CTL1 register,
TAA2IOC1 register,
TAA2OPT0 register
Initial setting of these registers is performed before setting the TAA2CE bit to 1.
TAA2CE bit = 1
The TAA2CKS0 to TAA2CKS2 bits can be set at the same time when counting has been started (TAA2CE bit = 1).
<2> Overflow flag clear flow
Read TAA2OPT0 register (check overflow flag).
TAA2OVF bit = 1
No
Yes
Execute instruction to clear TAA2OVF bit (CLR TAA2OVF).
<3> Count operation stop flow
TAA2CE bit = 0
Counter is initialized and counting is stopped by clearing TAA2CE bit to 0.
STOP

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter AA is used as an interval timer with the TAAnCCRa register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTAnCCa signal has been detected.


FFFFH
16-bit counter
0000H
D_{10}
D_{00}
D_{02}
D_{11}
D_{01}
D_{03}
D_{12}
D_{13}
D_{04}
TAAnCE bit
TAAnCCR0 register
D_{00}
D_{01}
D_{02}
D_{03}
D_{04}
D_{05}
INTTAnCC0 signal
TOAn0 pin output
Interval period (D_{00} + 1)
Interval period (10000H + D_{01} – D_{00})
Interval period (D_{02} – D_{01})
Interval period (10000H + D_{03} – D_{02})
Interval period (10000H + D_{04} – D_{03})
TAAnCCR1 register
D_{10}
D_{11}
D_{12}
D_{13}
D_{14}
INTTAnCC1 signal
TOAn1 pin output
Interval period (D_{10} + 1)
Interval period (10000H + D_{11} – D_{10})
Interval period (10000H + D_{12} – D_{11})
Interval period (10000H + D_{13} – D_{12})

When performing an interval operation in the free-running timer mode, two intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TAAnCCRa register must be re-set in the interrupt servicing that is executed when the INTTAnCCa signal is detected.

The set value for re-setting the TAAnCCRa register can be calculated by the following expression, where “D_a” is the interval period.

Compare register default value: D_a – 1

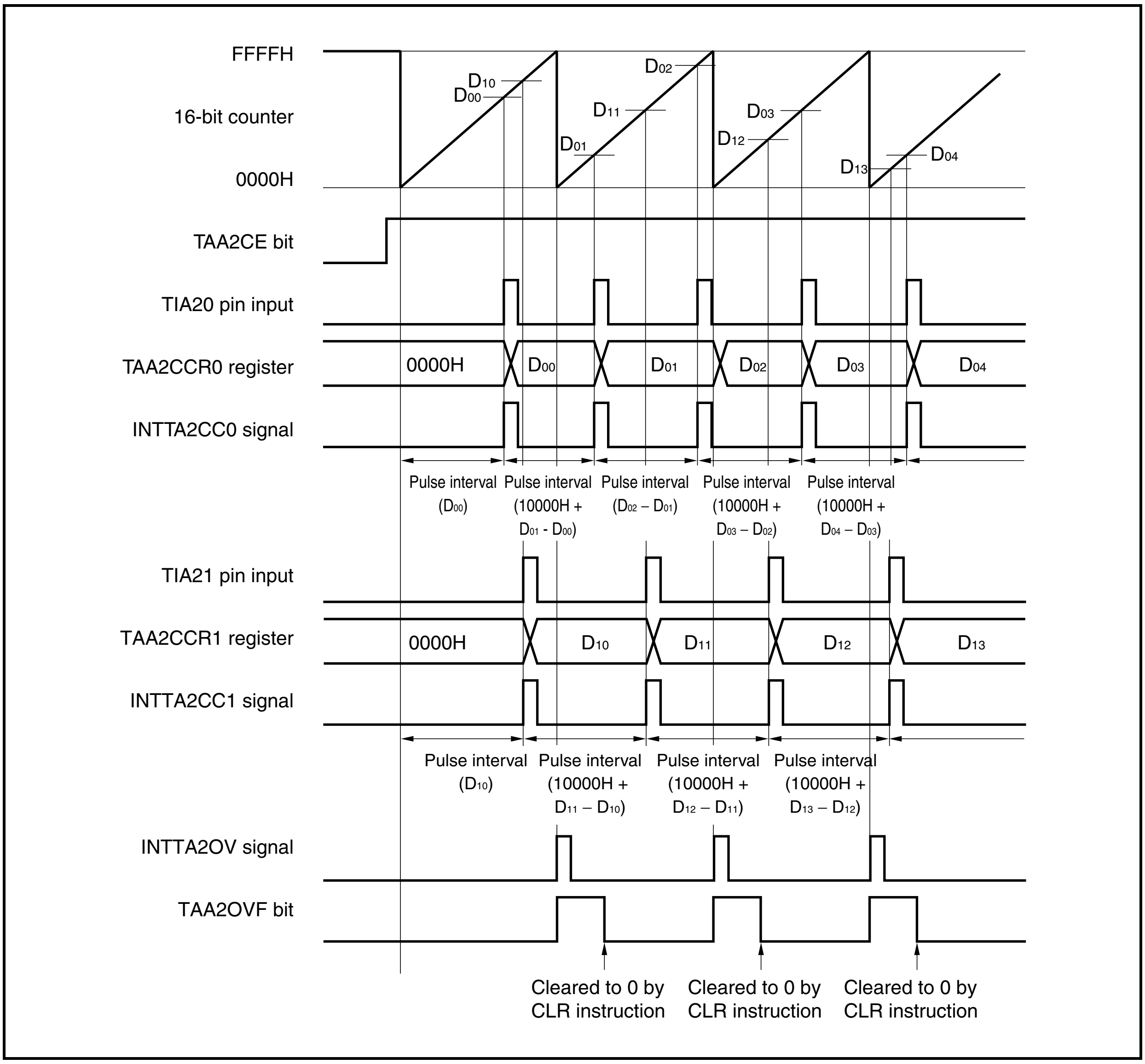
Value set to compare register second and subsequent time: Previous set value + D_a

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0 to 2
a = 0, 1

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TAA2CCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTA2CCa signal has been detected and for calculating an interval.


FFFFH
16-bit counter
0000H
D10
D00
D02
D11
D01
D03
D12
D13
D04
TAA2CE bit
TIA20 pin input
TAA2CCR0 register
0000H
D00
D01
D02
D03
D04
INTTA2CC0 signal
Pulse interval (D00)
Pulse interval (10000H + D01 - D00)
Pulse interval (D02 – D01)
Pulse interval (10000H + D03 – D02)
Pulse interval (10000H + D04 – D03)
TIA21 pin input
TAA2CCR1 register
0000H
D10
D11
D12
D13
INTTA2CC1 signal
Pulse interval (D10)
Pulse interval (10000H + D11 – D10)
Pulse interval (10000H + D12 – D11)
Pulse interval (10000H + D13 – D12)
INTTA2OV signal
TAA2OVF bit
Cleared to 0 by CLR instruction
Cleared to 0 by CLR instruction
Cleared to 0 by CLR instruction

When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

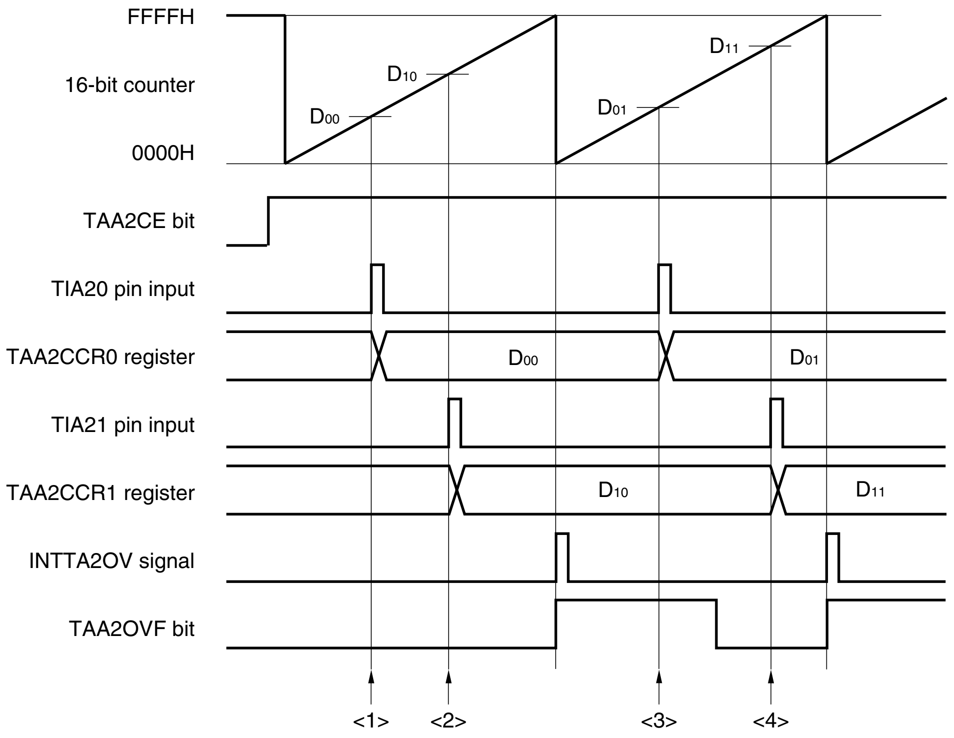
To measure a pulse width, the pulse width can be calculated by reading the value of the TAA2CCRa register in synchronization with the INTTA2CCa signal, and calculating the difference between the read value and the previously read value.

Remark a = 0, 1

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



The following problem may occur when two pulse widths are measured in the free-running timer mode.

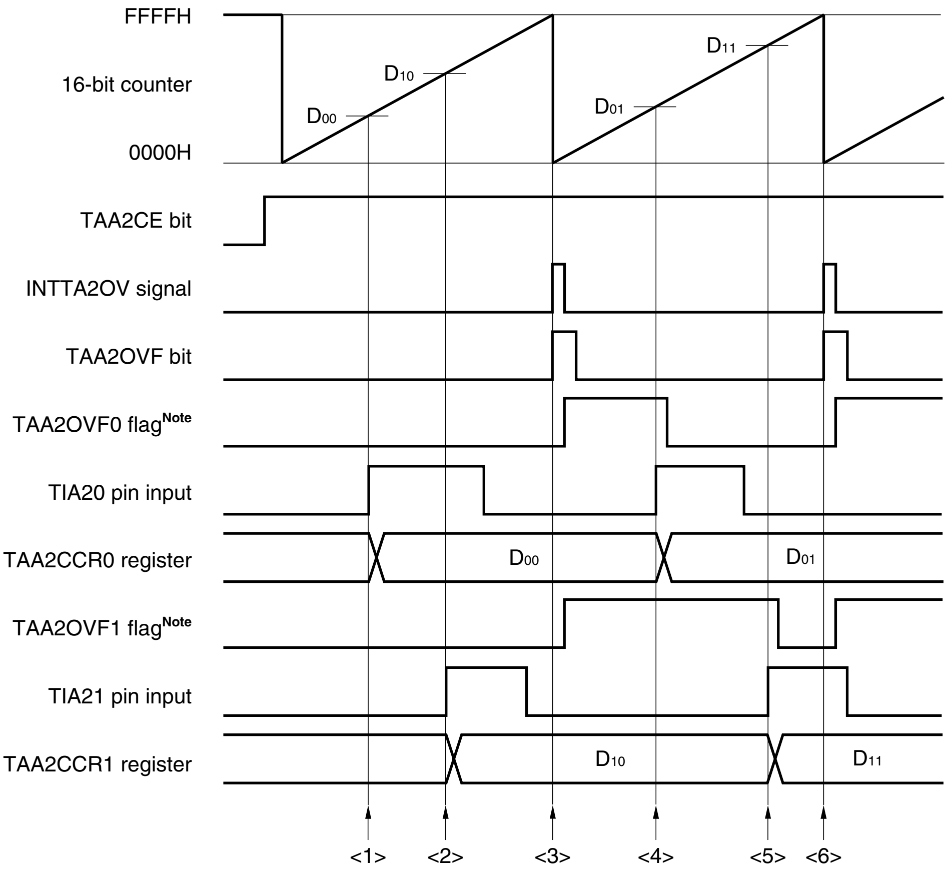
<1> Read the TAA2CCR0 register (setting of the default value of the TIA20 pin input).
<2> Read the TAA2CCR1 register (setting of the default value of the TIA21 pin input).
<3> Read the TAA2CCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<4> Read the TAA2CCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} − D_{10}) (incorrect).

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.

Use software when using two capture registers. An example of how to use software is shown below.

(1/2)

Example when two capture registers are used (using overflow interrupt)



Note The TAA2OVF0 and TAA2OVF1 flags are set on the internal RAM by software.

<1> Read the TAA2CCR0 register (setting of the default value of the TIA20 pin input).

<2> Read the TAA2CCR1 register (setting of the default value of the TIA21 pin input).

<3> An overflow occurs. Set the TAA2OVF0 and TAA2OVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

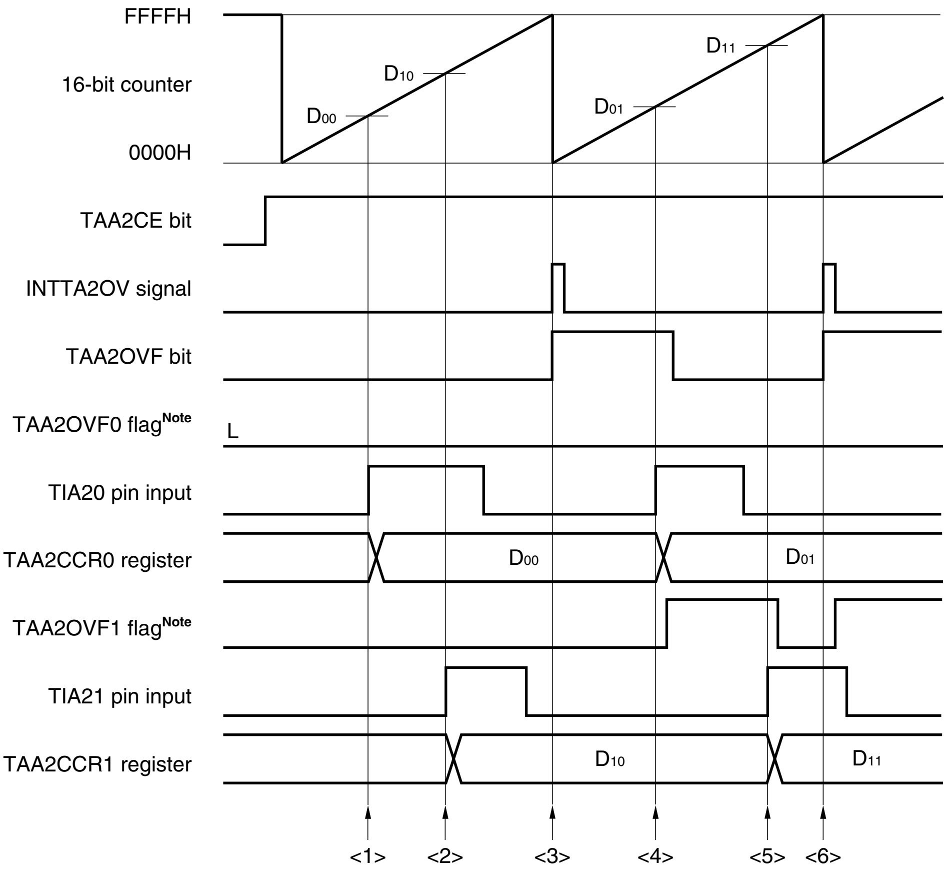
<4> Read the TAA2CCR0 register.
Read the TAA2OVF0 flag. If the TAA2OVF0 flag is 1, clear it to 0.
Because the TAA2OVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).

<5> Read the TAA2CCR1 register.
Read the TAA2OVF1 flag. If the TAA2OVF1 flag is 1, clear it to 0 (the TAA2OVF0 flag is cleared in <4>, and the TAA2OVF1 flag remains 1).
Because the TAA2OVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).

<6> Same as <3>

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TAA2OVF0 and TAA2OVF1 flags are set on the internal RAM by software.

<1> Read the TAA2CCR0 register (setting of the default value of the TIA20 pin input).

<2> Read the TAA2CCR1 register (setting of the default value of the TIA21 pin input).

<3> An overflow occurs. Nothing is done by software.

<4> Read the TAA2CCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TAA2OVF1 flag to 1, and clear the overflow flag to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).

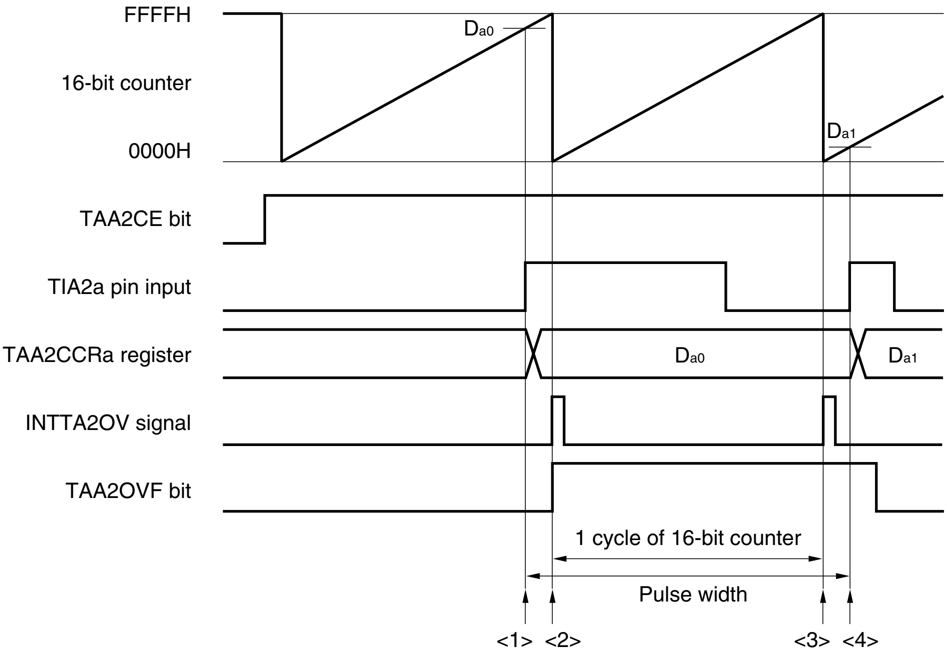
<5> Read the TAA2CCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TAA2OVF1 flag. If the TAA2OVF1 flag is 1, clear it to 0.
Because the TAA2OVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).

<6> Same as <3>

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



The following problem may occur when long pulse width is measured in the free-running timer mode.

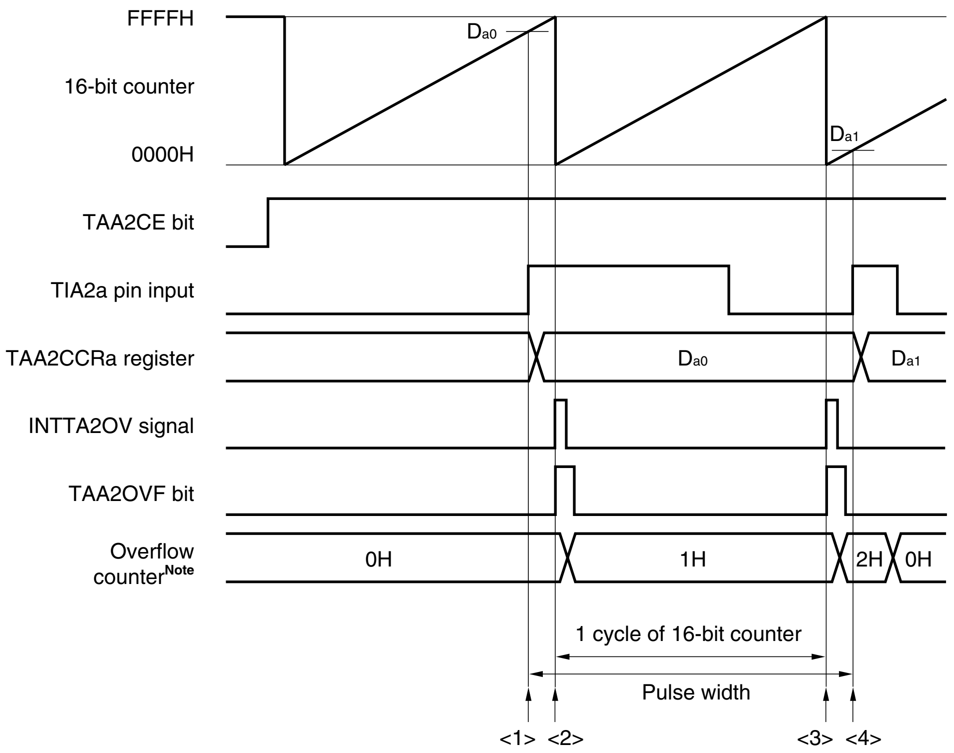
<1> Read the TAA2CCRa register (setting of the default value of the TIA2a pin input).
<2> An overflow occurs. Nothing is done by software.
<3> An overflow occurs a second time. Nothing is done by software.
<4> Read the TAA2CCRa register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{a1} − D_{a0}) (incorrect).
Actually, the pulse width must be (20000H + D_{a1} − D_{a0}) because an overflow occurs twice.

Remark a = 0, 1

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.

If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TAA2CCRa register (setting of the default value of the TIA2a pin input).
<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.
<3> An overflow occurs a second time. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.
<4> Read the TAA2CCRa register.
Read the overflow counter.
→ When the overflow counter is "N", the pulse width can be calculated by (N × 10000H + D_{a1} – D_{a0}).
In this example, the pulse width is (20000H + D_{a1} – D_{a0}) because an overflow occurs twice.
Clear the overflow counter (0H).

Remark a = 0, 1

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TAA2OVF bit to 0 with the CLR instruction after reading the TAA2OVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TAA2OPT0 register after reading the TAA2OVF bit when it is 1.

6.6.7 Pulse width measurement mode (TAA2MD2 to TAA2MD0 bits = 110)

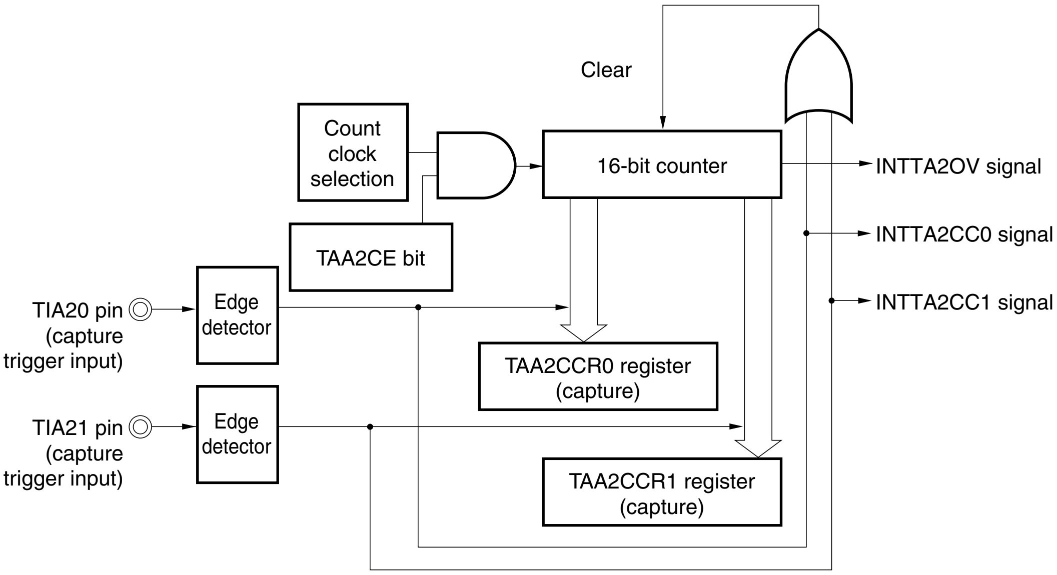
This mode is valid only in TAA2.

In the pulse width measurement mode, 16-bit timer/event counter AA starts counting when the TAA2CTL0.TAA2CE bit is set to 1. Each time the valid edge input to the TIA2a pin has been detected, the count value of the 16-bit counter is stored in the TAA2CCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TAA2CCRa register after a capture interrupt request signal (INTTA2CCa) occurs.

As shown in Figure 6-45, select either the TIA20 or TIA21 pin as the capture trigger input pin and set the unused pins to “No edge detection” by using the TAA2IOC1 register.

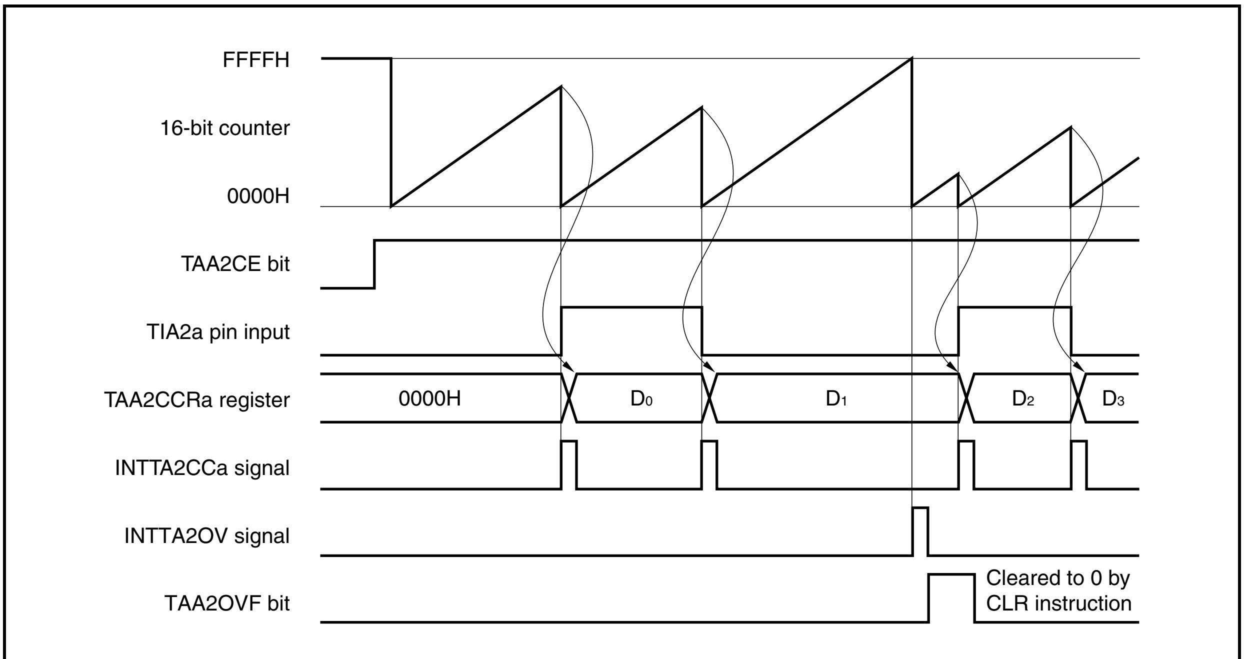
Figure 6-44. Configuration in Pulse Width Measurement Mode



Caution In the pulse width measurement mode, select the internal clock as the count clock (by clearing the TAA2CTL1.TAA2EEE bit to 0).

Remark a = 0, 1

Figure 6-45. Basic Timing in Pulse Width Measurement Mode



Remark a = 0, 1

When the TAA2CE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIA2a pin is later detected, the count value of the 16-bit counter is stored in the TAA2CCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTA2CCa) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

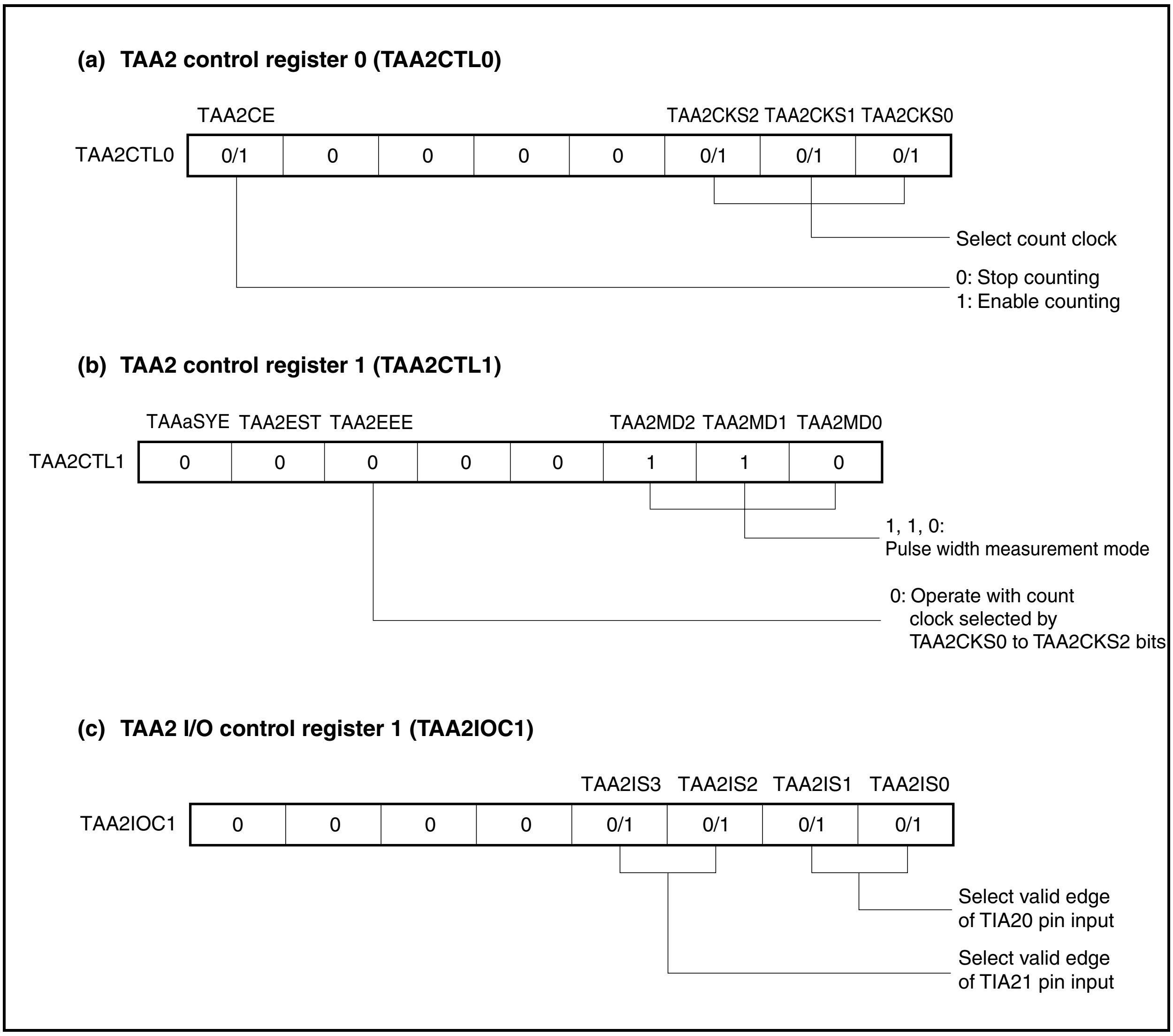
If the valid edge is not input to the TIA2a pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTA2OV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TAA2OPT0.TAA2OVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TAA2OVF bit set (1) count + Captured value) × Count clock cycle

Remark a = 0, 1

Figure 6-46. Register Setting in Pulse Width Measurement Mode (1/2)



(a) TAA2 control register 0 (TAA2CTL0)

	TAA2CE					TAA2CKS2	TAA2CKS1	TAA2CKS0
TAA2CTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TAA2 control register 1 (TAA2CTL1)

	TAAaSYE	TAA2EST	TAA2EEE			TAA2MD2	TAA2MD1	TAA2MD0
TAA2CTL1	0	0	0	0	0	1	1	0

1, 1, 0:
Pulse width measurement mode

0: Operate with count
clock selected by
TAA2CKS0 to TAA2CKS2 bits

(c) TAA2 I/O control register 1 (TAA2IOC1)

					TAA2IS3	TAA2IS2	TAA2IS1	TAA2IS0
TAA2IOC1	0	0	0	0	0/1	0/1	0/1	0/1

Select valid edge
of TIA20 pin input

Select valid edge
of TIA21 pin input

Figure 6-46. Register Setting in Pulse Width Measurement Mode (2/2)

(d) TAA2 option register 0 (TAA2OPT0)

			TAA2CCS1	TAA2CCS0				TAA2OVF
TAA2OPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(e) TAA2 counter read buffer register (TAA2CNT)

The value of the 16-bit counter can be read by reading the TAA2CNT register.

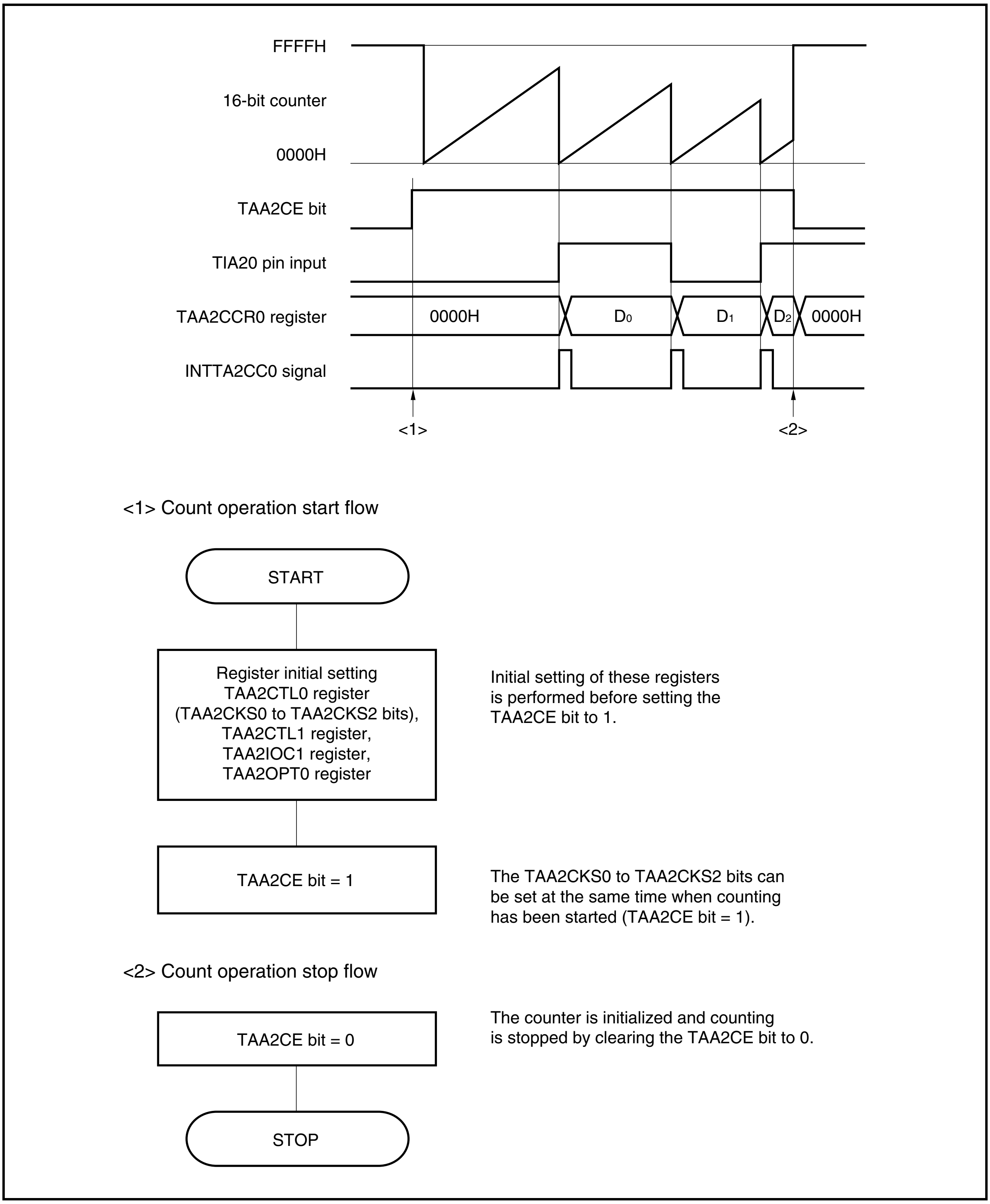
(f) TAA2 capture/compare registers 0 and 1 (TAA2CCR0 and TAA2CCR1)

These registers store the count value of the 16-bit counter when the valid edge input to the TIA20 and TIA21 pins is detected.

Remark TAA2 I/O control register 0 (TAA2IOC0) and TAA2 I/O control register 2 (TAA2IOC2) are not used in the pulse width measurement mode.

(1) Operation flow in pulse width measurement mode

Figure 6-47. Software Processing Flow in Pulse Width Measurement Mode



Initial setting of these registers is performed before setting the TAA2CE bit to 1.

The TAA2CKS0 to TAA2CKS2 bits can be set at the same time when counting has been started (TAA2CE bit = 1).

The counter is initialized and counting is stopped by clearing the TAA2CE bit to 0.

(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TAA2OVF bit to 0 with the CLR instruction after reading the TAA2OVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TAA2OPT0 register after reading the TAA2OVF bit when it is 1.

CHAPTER 7 16-BIT TIMER/EVENT COUNTER AB (TAB)

Timer AB (TAB) is a 16-bit timer/event counter.

The V850E/IG4-H and V850E/IH4-H incorporate TAB0 and TAB1.

7.1 Overview

7.1.1 TAB0 of V850E/IG4-H, and TAB0 and TAB1 of V850E/IH4-H

An outline of TAB0 of the V850E/IG4-H, and TAB0 and TAB1 of the V850E/IH4-H is shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: 4
- External event count input pins: 1
- External trigger input pins: 1
- Timer/counters: 1
- Capture/compare registers: 4
- Capture/compare match interrupt request signals: 4
- Overflow interrupt request signal: 1
- Timer output pins[Note]: 4

Note This is the number of output pins of TABn; it does not include the output pins of TMQOPn. For details of the output pins of TMQOPn, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

7.1.2 TAB1 of V850E/IG4-H

An outline of TAB1 of the V850E/IG4-H is shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: None
- External event count input pins: 1
- External trigger input pins: 1
- Timer/counters: 1
- Capture/compare registers: 4
- Capture/compare match interrupt request signals: 4
- Overflow interrupt request signal: 1
- Timer output pins[Note]: 1

Note This is the number of output pins of TAB1.

7.2 Functions

7.2.1 TAB0 of V850E/IG4-H, and TAB0 and TAB1 of V850E/IH4-H

TAB0 of the V850E/IG4-H, and TAB0 and TAB1 of the V850E/IH4-H have the following functions.

- 6-phase PWM output[Note]
- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement

Note This is connected to TMQOPn. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

7.2.2 TAB1 of V850E/IG4-H

TAB1 of the V850E/IG4-H has the following functions.

- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer (compare function only)

7.3 Configuration

7.3.1 TAB0 of V850E/IG4-H, and TAB0 and TAB1 of V850E/IH4-H

TAB0 of the V850E/IG4-H, and TAB0 and TAB1 of the V850E/IH4-H include the following hardware.

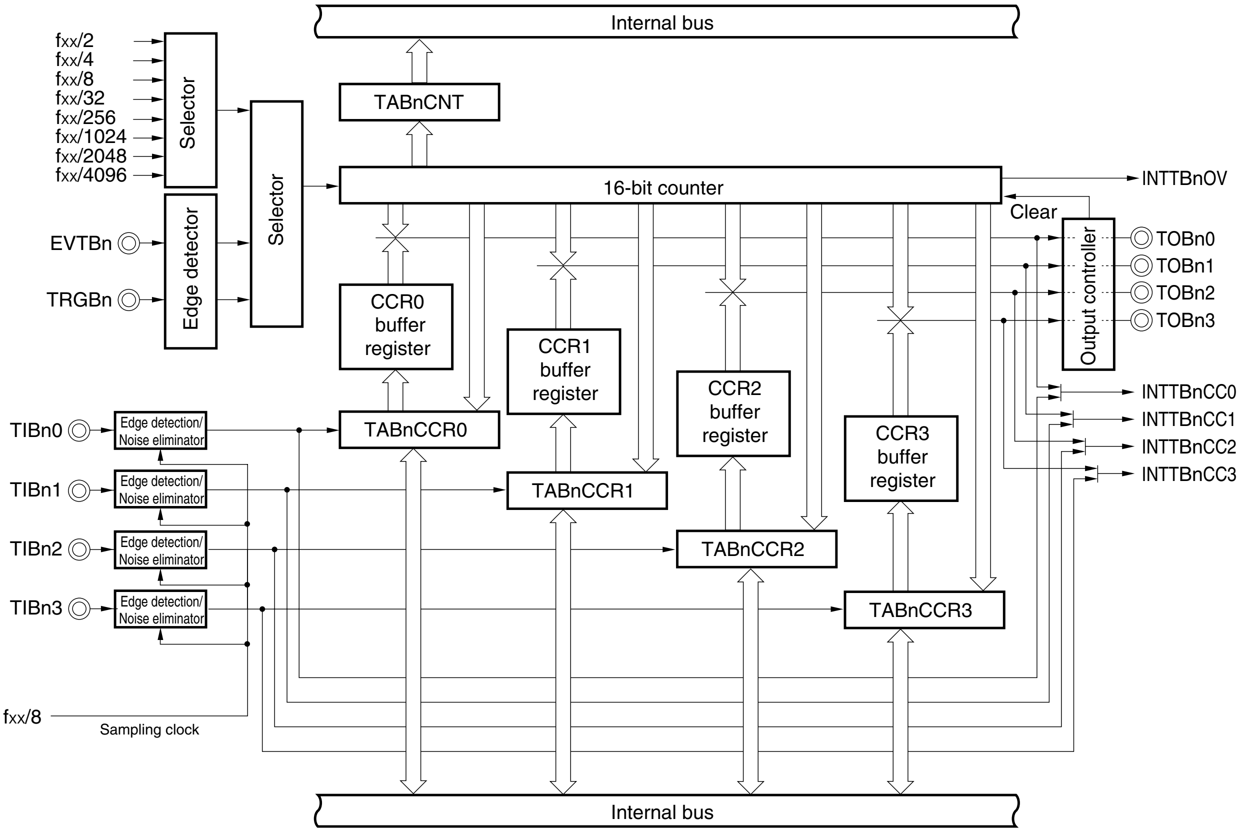
Table 7-1. Configuration of TAB0 of V850E/IG4-H, and TAB0 and TAB1 of V850E/IH4-H

Item	Configuration
Timer register	16-bit counter × 1
Registers	TABn counter read buffer register (TABnCNT) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3) CCR0 to CCR3 buffer registers
Timer input	12 in total (TIB00 to TIB03, TIB10 to TIB13, EVTB0, EVTB1, TRGB0, TRGB1 pins)[Note]
Timer output	8 in total (TOB00 to TOB03, TOB10 to TOB13 pins)[Note]
Control registers	TABn control registers 0, 1 (TABnCTL0, TABnCTL1) TABn I/O control registers 0 to 2 (TABnIOC0 to TABnIOC2) TABn option register 0 (TABnOPT0)

Note The TIBn1 to TIBn3 pins function alternately as timer output pins (TOBn1 to TOBn3).

Remark n = 0, 1

Figure 7-1. Block Diagram of TAB0 of V850E/IG4-H, and TAB0 and TAB1 of V850E/IH4-H



Remarks 1. fxx: Peripheral clock

2. For the noise eliminator, see **4.6 Noise Eliminator**.

3. n = 0, 1

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TABnCNT register.

When the TABnCTL0.TABnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TABnCNT register is read at this time, 0000H is read.

Reset sets the TABnCE bit to 0.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TABnCCR0 register is used as a compare register, the value written to the TABnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTBnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TABnCCR0 register is cleared to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TABnCCR1 register is used as a compare register, the value written to the TABnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTBnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, and the TABnCCR1 register is cleared to 0000H.

(4) CCR2 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TABnCCR2 register is used as a compare register, the value written to the TABnCCR2 register is transferred to the CCR2 buffer register. When the count value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTBnCC2) is generated.

The CCR2 buffer register cannot be read or written directly.

The CCR2 buffer register is cleared to 0000H after reset, and the TABnCCR2 register is cleared to 0000H.

(5) CCR3 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TABnCCR3 register is used as a compare register, the value written to the TABnCCR3 register is transferred to the CCR3 buffer register. When the count value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTBnCC3) is generated.

The CCR3 buffer register cannot be read or written directly.

The CCR3 buffer register is cleared to 0000H after reset, and the TABnCCR3 register is cleared to 0000H.

(6) Edge detector

This circuit detects the valid edges input to the TIBn0 to TIBn3, EVTBn, and TRGBn pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TABnIOC1 and TABnIOC2 registers.

(7) Output controller

This circuit controls the output of the TOBn0 to TOBn3 pins. The output controller is controlled by the TABnIOC0 register.

(8) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

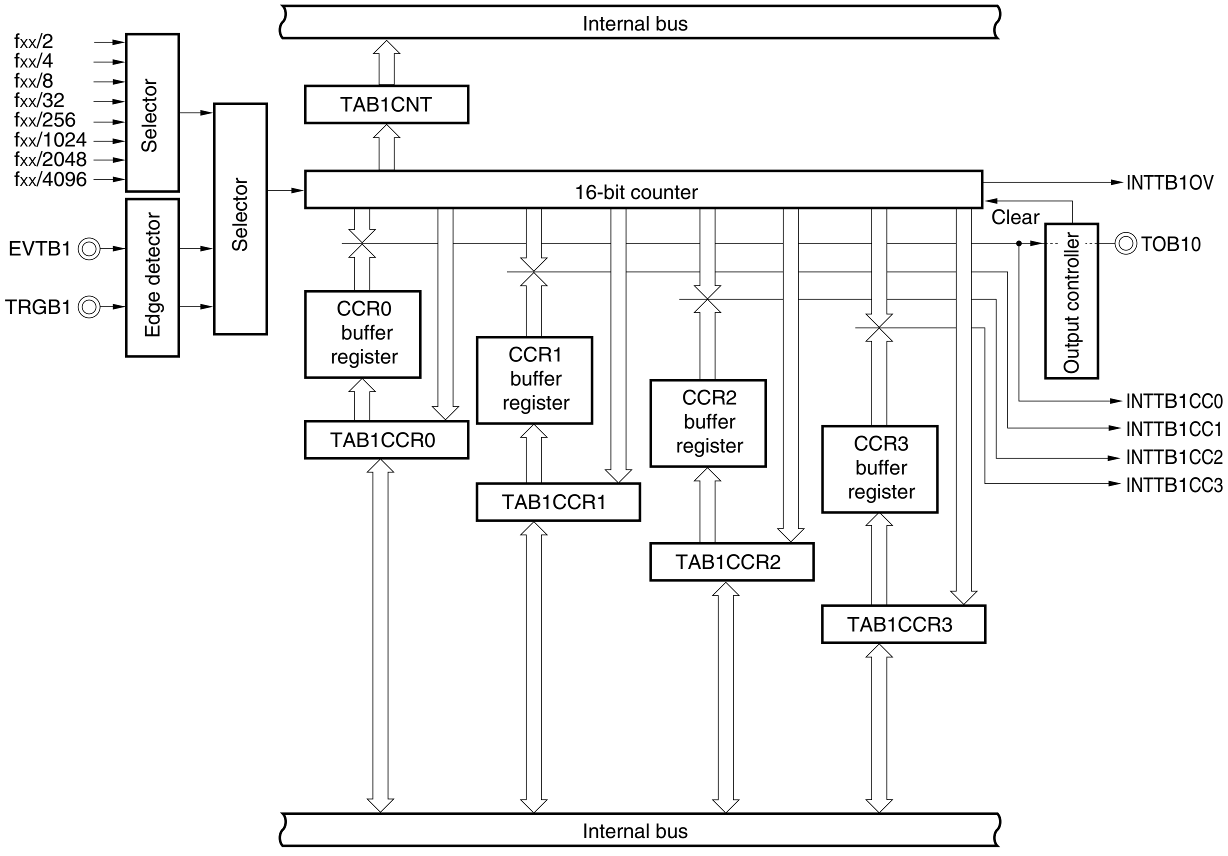
7.3.2 TAB1 of V850E/IG4-H

TAB1 of the V850E/IG4-H includes the following hardware.

Table 7-2. Configuration of TAB1 of V850E/IG4-H

Item	Configuration
Timer register	16-bit counter × 1
Registers	TAB1 counter read buffer register (TAB1CNT) TAB1 capture/compare registers 0 to 3 (TAB1CCR0 to TAB1CCR3) CCR0 to CCR3 buffer registers
Timer input	4 in total (EVTB0, EVTB1, TRGB0, TRGB1 pins)
Timer output	1 in total (TOB10 pin)
Control registers	TAB1 control registers 0, 1 (TAB1CTL0, TAB1CTL1) TAB1 I/O control registers 0 to 2 (TAB1IOC0 to TAB1IOC2) TAB1 option register 0 (TAB1OPT0)

Figure 7-2. Block Diagram of TAB1 of V850E/IG4-H



Remark fxx: Peripheral clock

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TAB1CNT register.

When the TAB1CTL0.TAB1CE bit = 0, the value of the 16-bit counter is FFFFH. If the TAB1CNT register is read at this time, 0000H is read.

Reset sets the TAB1CE bit to 0.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TAB1CCR0 register is used as a compare register, the value written to the TAB1CCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTB1CC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TAB1CCR0 register is cleared to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TAB1CCR1 register is used as a compare register, the value written to the TAB1CCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTB1CC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, and the TAB1CCR1 register is cleared to 0000H.

(4) CCR2 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TAB1CCR2 register is used as a compare register, the value written to the TAB1CCR2 register is transferred to the CCR2 buffer register. When the count value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTB1CC2) is generated.

The CCR2 buffer register cannot be read or written directly.

The CCR2 buffer register is cleared to 0000H after reset, and the TAB1CCR2 register is cleared to 0000H.

(5) CCR3 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TAB1CCR3 register is used as a compare register, the value written to the TAB1CCR3 register is transferred to the CCR3 buffer register. When the count value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTB1CC3) is generated.

The CCR3 buffer register cannot be read or written directly.

The CCR3 buffer register is cleared to 0000H after reset, and the TAB1CCR3 register is cleared to 0000H.

(6) Edge detector

This circuit detects the valid edges input to the EVTB1 and TRGB1 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TAB1IOC2 register.

(7) Output controller

This circuit controls the output of the TOB10 pin. The output controller is controlled by the TAB1IOC0 register.

(8) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

7.4 Registers

(1) TABn control register 0 (TABnCTL0)

The TABnCTL0 register is an 8-bit register that controls the operation of TABn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

The same value can always be written to the TABnCTL0 register by software.

After reset: 00H R/W Address: TAB0CTL0 FFFFF5E0H, TAB1CTL0 FFFFF620H

	<7>	6	5	4	3	2	1	0
TABnCTL0	TABnCE	0	0	0	0	TABnCKS2	TABnCKS1	TABnCKS0

V850E/IG4-H
n = 0, 1
m = 0

V850E/IH4-H
n = 0, 1
m = 0, 1

TABnCE	TABn operation control
0	TABn operation disabled (TABn reset asynchronously[Note])
1	TABn operation enabled.

TABnCKS2	TABnCKS1	TABnCKS0	Internal count clock selection
0	0	0	fxx/2
0	0	1	fxx/4
0	1	0	fxx/8
0	1	1	fxx/32
1	0	0	fxx/256
1	0	1	fxx/1024
1	1	0	fxx/2048
1	1	1	fxx/4096

Note The TABnOPT0.TABnOVF bit and the 16-bit counter are reset simultaneously. Moreover, timer outputs (TOBn0, TOBm1 to TOBm3 pins) are reset to the TABnIOC0 register set status at the same time as the 16-bit counter is reset.

Cautions 1. Set the TABnCKS2 to TABnCKS0 bits when the TABnCE bit = 0.
When the value of the TABnCE bit is changed from 0 to 1, the TABnCKS2 to TABnCKS0 bits can be set simultaneously.

2. Be sure to set bits 3 to 6 to “0”.

Remark fxx: Peripheral clock

(2) TABn control register 1 (TABnCTL1)

The TABnCTL1 register is an 8-bit register that controls the operation of TABn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0CTL1 FFFFF5E1H, TAB1CTL1 FFFFF621H

	7	6	5	4	3	2	1	0
TABnCTL1 (n = 0, 1)	0	TABnEST	TABnEEE	0	0	TABnMD2	TABnMD1	TABnMD0

TABnEST	Software trigger control
0	–
1	Generate a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TABnEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TABnEST bit as the trigger.
Read value of the TABnEST bit is always 0.	

TABnEEE	Count clock selection
0	Disable operation with external event count input (EVTBn pin). (Perform counting with the count clock selected by the TABnCTL0.TABnCKS0 to TABnCKS2 bits.)
1	Enable operation with external event count input (EVTBn pin). (Perform counting at the valid edge of the external event count input signal (EVTBn pin).)
The TABnEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TABnMD2	TABnMD1	TABnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode[Note 1]
1	1	1	6-phase PWM output mode[Notes 1, 2]

Notes 1. For the V850E/IG4-H, only TAB0 can be set. Setting TAB1 is prohibited.

2. The 6-phase PWM output mode cannot be used when only TABn is used. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

Cautions 1. The TABnEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. External event count input is selected in the external event count mode regardless of the value of the TABnEEE bit.

3. Set the TABnEEE and TABnMD2 to TABnMD0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TABnCE bit = 1. If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

4. Be sure to set bits 3, 4, and 7 to "0".

(3) TABn I/O control register 0 (TABnIOC0)

The TABnIOC0 register is an 8-bit register that controls the timer output (the TOBn0 to TOBn3, and TOBnT1 to TOBnT3 pins (the TOB11 to TOB13, and TOB1T1 and TOB1T2 pins are available only in the V850E/IH4-H)).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(a) TAB0 of V850E/IG4-H, and TAB0 and TAB1 of V850E/IH4-H

After reset: 00H R/W Address: TAB0IOC0 FFFFF5E2H, TAB1IOC0 FFFFF622H

	7	<6>	5	<4>	3	<2>	1	<0>
TABnIOC0	TABnOL3	TABnOE3	TABnOL2	TABnOE2	TABnOL1	TABnOE1	TABnOL0	TABnOE0

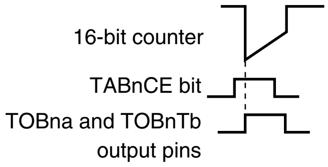
n = 0, 1
a = 0 to 3
b = 1 to 3

TABnOLa	TOBna, TOBnTb pin output level setting[Note]
0	TOBna and TOBnTb pins start output at high level.
1	TOBna and TOBnTb pins start output at low level.

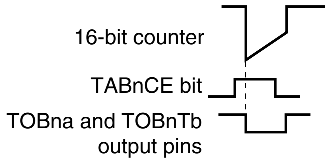
TABnOEa	TOBna, TOBnTb pin output setting
0	Timer output disabled • When TABnOLa bit = 0: Low level is output from the TOBna and TOBnTb pins • When TABnOLa bit = 1: High level is output from the TOBna and TOBnTb pins
1	Timer output enabled (A pulse is output from the TOBna and TOBnTb pins).

Note The output level of the timer output pins (TOBna and TOBnTb) specified by the TABnOLa bit is shown below.

- When TABnOLa bit = 0



- When TABnOLa bit = 1



Cautions 1. If the setting of the TABnIOC0 register is changed when TOBna and TOBnTb are set in the output mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.

2. Rewrite the TABnOLa and TABnOEa bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear (0) the TABnCE bit and then set the bits again.

3. If the TABnOLa bit is manipulated when the TABnCE and TABnOEa bits are 0, the output level of the TOBna and TOBnTb pins changes.

4. To generate the TOBnTb pin output and the A/D conversion start trigger signal of A/D converters 0 and 1 in the 6-phase PWM output mode, be sure to set the TOBnTb pin output using the TABnIOC0 register. At this time, be sure to clear the TABnOL0 bit to 0 and set the TABnOE0 bit to 1.

(b) TAB1 of V850E/IG4-H

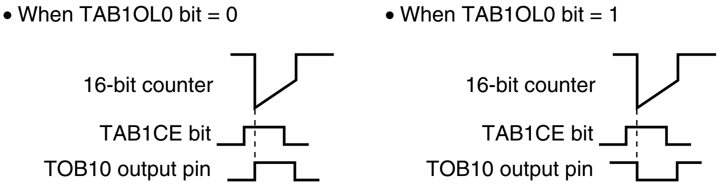
After reset: 00H R/W Address: FFFFF622H

	7	6	5	4	3	2	1	<0>
TAB1IOC0	0	0	0	0	0	0	TAB1OL0	TAB1OE0

TAB1OL0	TOB10 pin output level setting[Note]
0	TOB10 pin starts output at high level.
1	TOB10 pin starts output at low level.

TAB1OE0	TOB10 pin output setting
0	Timer output disabled • When TAB1OL0 bit = 0: Low level is output from the TOB10 pin • When TAB1OL0 bit = 1: High level is output from the TOB10 pin
1	Timer output enabled (A pulse is output from the TOB10 pin).

Note The output level of the timer output pin (TOB10) specified by the TAB1OL0 bit is shown below.



Cautions 1. If the setting of the TAB1IOC0 register is changed when TOB10 is set in the output mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.

2. Rewrite the TAB1OL0 and TAB1OE0 bits when the TAB1CTL0.TAB1CE bit = 0. (The same value can be written when the TAB1CE bit = 1.) If rewriting was mistakenly performed, clear (0) the TAB1CE bit and then set the bits again.

3. If the TAB1OL0 bit is manipulated when the TAB1CE and TAB1OE0 bits are 0, the output level of the TOB10 pin changes.

(4) TABm I/O control register 1 (TABmIOC1)

The TABmIOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (TIBm0 to TIBm3 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0IOC1 FFFFF5E3H, TAB1IOC1 FFFFF623H[Note]

	7	6	5	4	3	2	1	0
TABmIOC1	TABmIS7	TABmIS6	TABmIS5	TABmIS4	TABmIS3	TABmIS2	TABmIS1	TABmIS0

V850E/IG4-H m = 0

V850E/IH4-H m = 0, 1

TABmIS7	TABmIS6	Capture trigger input signal (TIBm3 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABmIS5	TABmIS4	Capture trigger input signal (TIBm2 pin) valid edge detection
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABmIS3	TABmIS2	Capture trigger input signal (TIBm1 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABmIS1	TABmIS0	Capture trigger input signal (TIBm0 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Note V850E/IH4-H only

Cautions 1. Rewrite the TABmIS7 to TABmIS0 bits when the TABmCTL0.TABmCE bit = 0. (The same value can be written when the TABmCE bit = 1.) If rewriting was mistakenly performed, clear the TABmCE bit to 0 and then set the bits again.

2. The TABmIS7 to TABmIS0 bits are valid only in the free-running timer mode (only when the TABmOPT0.TABmCCSa bit = 1) and the pulse width measurement mode (a = 0 to 3). In all other modes, a capture operation is not possible.

(5) TABn I/O control register 2 (TABnIOC2)

The TABnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (EVTBn pin) and external trigger input signal (TRGBn pin).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0IOC2 FFFFF5E4H, TAB1IOC2 FFFFF624H

	7	6	5	4	3	2	1	0
TABnIOC2 (n = 0, 1)	0	0	0	0	TABnEES1	TABnEES0	TABnETS1	TABnETS0

TABnEES1	TABnEES0	External event count input signal (EVTBn pin) valid edge setting
0	0	No edge detection (external event count invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnETS1	TABnETS0	External trigger input signal (TRGBn pin) valid edge setting
0	0	No edge detection (external trigger invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TABnEES1, TABnEES0, TABnETS1, and TABnETS0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. The TABnEES1 and TABnEES0 bits are valid only when the TABnCTL1.TABnEEE bit = 1 or when the external event count mode (TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 001) has been set.

3. The TABnETS1 and TABnETS0 bits are valid only in the external trigger pulse output mode or one-shot pulse output mode.

(6) TABn option register 0 (TABnOPT0)

The TABnOPT0 register is an 8-bit register used to set the capture/compare operation and detect an overflow.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0OPT0 FFFFF5E5H, TAB1OPT0 FFFFF625H

	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TABnOPT0	TABmCCS3[Note 1]	TABmCCS2[Note 1]	TABmCCS1[Note 1]	TABmCCS0[Note 1]	0	TABnCMS[Note 2]	TABnCUF[Note 2]	TABnOVF

V850E/IG4-H
n = 0, 1
m = 0
a = 0 to 3

V850E/IH4-H
n = 0, 1
m = 0, 1
a = 0 to 3

TABmCCSa[Note 1]	TABmCCRa register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TABmCTL0.TABmCE bit = 0)
The TABmCCSa bit setting is valid only in the free-running timer mode.	

TABnOVF	TABn overflow flag
Set (1)	Overflow occurred
Reset (0)	TABnOVF bit 0 written or TABnCTL0.TABnCE bit = 0
• The TABnOVF bit is set to 1 when the 16-bit counter count value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode[Note 3]. • An overflow interrupt request signal (INTTBnOV) is generated at the same time that the TABnOVF bit is set to 1. The INTTBnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode[Note 3]. • The TABnOVF bit is not cleared to 0 even when the TABnOVF bit or the TABnOPT0 register are read when the TABnOVF bit = 1. • Before clearing the TABnOVF bit to 0 after generation of the INTTBnOV signal, be sure to confirm (by reading) that the TABnOVF bit is set to 1. • The TABnOVF bit can be both read and written, but the TABnOVF bit cannot be set to 1 by software. Writing 1 has no influence on the operation of TABn.	

Notes 1. For the V850E/IG4-H, only TAB0 can be set. Be sure to set bits 4 to 7 of TAB1 to 0.

2. Be sure to set bits 1 and 2 of TAB1 to 0 for the V850E/IG4-H.

For details of the TABnCMS and TABnCUF bits, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

3. In the free-running mode or the pulse width measurement mode, both TAB0 and TAB1 can be used in the V850E/IH4-H, but only TAB0 can be used in the V850E/IG4-H. .

Cautions 1. Rewrite the TABnCCS3 to TABnCCS0 bits when the TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. Be sure to set bit 3 to “0”.

(7) TABn capture/compare register 0 (TABnCCR0)

The TABmCCR0 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode. The TAB1CCR0 register of the V850E/IG4-H is a 16-bit register that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABmOPT0.TABmCCS0 bit. In the pulse width measurement mode, the TABnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR0 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

After reset: 0000H R/W Address: TAB0CCR0 FFFFF5E6H, TAB1CCR0 FFFFF626H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR0																
(n = 0, 1)																

(a) Function as compare register

The TABnCCR0 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTBnCC0) is generated. If TOBn0 pin output is enabled at this time, the output of the TOBn0 pin is inverted.

When the TABnCCR0 register is used as a cycle register in the interval timer mode, external event count mode, external trigger pulse output mode, one-shot pulse output mode, or PWM output mode, the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

(b) Function as capture register

When the TABmCCR0 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABmCCR0 register if the valid edge of the capture trigger input pin (TIBm0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABmCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBm0 pin) is detected.

Even if the capture operation and reading the TABmCCR0 register conflict, the correct value of the TABmCCR0 register can be read.

The capture register is cleared by setting the TABmCTL0.TABmCE bit = 0.

Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note 2]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note 2]
Free-running timer	Capture[Note 1]/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. Both TAB0 and TAB1 can be used in the V850E/IH4-H, but only TAB0 can be used in the V850E/IG4-H.

2. Writing to the TABnCCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(8) TABn capture/compare register 1 (TABnCCR1)

The TABmCCR1 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode. The TAB1CCR1 register of the V850E/IG4-H is a 16-bit register that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABmOPT0.TABmCCS1 bit. In the pulse width measurement mode, the TABmCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR1 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

After reset: 0000H R/W Address: TAB0CCR1 FFFFF5E8H, TAB1CCR1 FFFFF628H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR1 (n = 0, 1)																

(a) Function as compare register

The TABnCCR1 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTBnCC1) is generated. If TOBm1 pin output is enabled at this time, the output of the TOBm1 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

(b) Function as capture register

When the TABmCCR1 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABmCCR1 register if the valid edge of the capture trigger input pin (TIBm1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABmCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBm1 pin) is detected.

Even if the capture operation and reading the TABmCCR1 register conflict, the correct value of the TABmCCR1 register can be read.

The capture register is cleared by setting the TABmCTL0.TABmCE bit to 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note 2]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note 2]
Free-running timer	Capture[Note 1]/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. Both TAB0 and TAB1 can be used in the V850E/IH4-H, but only TAB0 can be used in the V850E/IG4-H.

2. Writing to the TABnCCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(9) TABn capture/compare register 2 (TABnCCR2)

The TABmCCR2 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode. The TAB1CCR2 register of V850E/IG4-H is a 16-bit register that can be only used as a compare register.

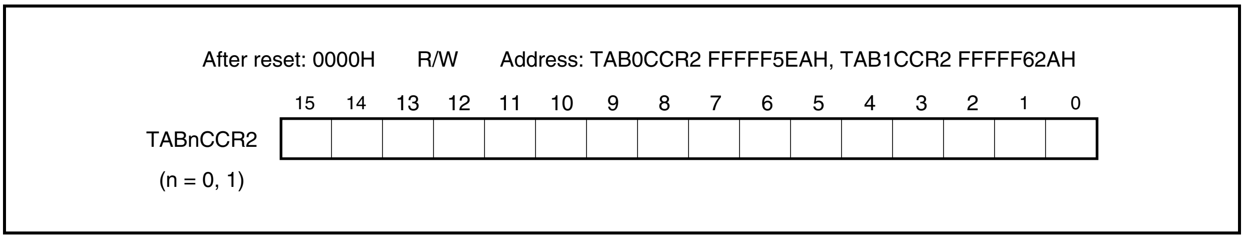
This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABmOPT0.TABmCCS2 bit. In the pulse width measurement mode, the TABmCCR2 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR2 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1



After reset: 0000H R/W Address: TAB0CCR2 FFFFF5EAH, TAB1CCR2 FFFFF62AH

(a) Function as compare register

The TABnCCR2 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR2 register is transferred to the CCR2 buffer register. When the value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTBnCC2) is generated. If TOBm2 pin output is enabled at this time, the output of the TOBm2 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

(b) Function as capture register

When the TABmCCR2 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABmCCR2 register if the valid edge of the capture trigger input pin (TIBm2 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABmCCR2 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBm2 pin) is detected.

Even if the capture operation and reading the TABmCCR2 register conflict, the correct value of the TABmCCR2 register can be read.

The capture register is cleared by setting the TABmCTL0.TABmCE bit to 0.

Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note 2]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note 2]
Free-running timer	Capture[Note 1]/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. Both TAB0 and TAB1 can be used in the V850E/IH4-H, but only TAB0 can be used in the V850E/IG4-H.

2. Writing to the TABnCCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(10) TABn capture/compare register 3 (TABnCCR3)

The TABmCCR3 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode. The TAB1CCR3 register of the V850E/IG4-H is a 16-bit register that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABmOPT0.TABmCCS3 bit. In the pulse width measurement mode, the TABmCCR3 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR3 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

After reset: 0000H R/W Address: TAB0CCR3 FFFFF5ECH, TAB1CCR3 FFFFF62CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR3																
(n = 0, 1)																

(a) Function as compare register

The TABnCCR3 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR3 register is transferred to the CCR3 buffer register. When the value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTBnCC3) is generated. If TOBm3 pin output is enabled at this time, the output of the TOBm3 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

(b) Function as capture register

When the TABmCCR3 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABmCCR3 register if the valid edge of the capture trigger input pin (TIBm3 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABmCCR3 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBm3 pin) is detected.

Even if the capture operation and reading the TABmCCR3 register conflict, the correct value of the TABmCCR3 register can be read.

The capture register is cleared by setting the TABmCTL0.TABmCE bit to 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-6. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note 2]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note 2]
Free-running timer	Capture[Note 1]/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. Both TAB0 and TAB1 can be used in the V850E/IH4-H, but only TAB0 can be used in the V850E/IG4-H.

2. Writing to the TABnCCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(11) TABn counter read buffer register (TABnCNT)

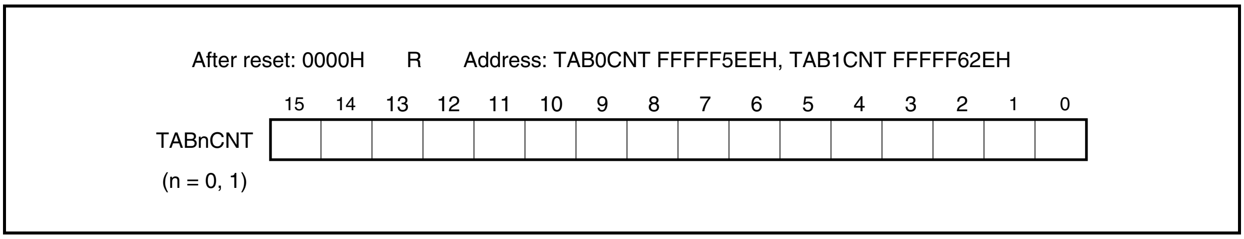
The TABnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TABnCTL0.TABnCE bit = 1, the count value of the 16-bit timer can be read.

This register is read-only, in 16-bit units.

The value of the TABnCNT register is set to 0000H when the TABnCE bit = 0. If the TABnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The value of the TABnCNT register is set to 0000H after reset, and the TABnCE bit is cleared to 0.



After reset: 0000H R Address: TAB0CNT FFFFF5EEH, TAB1CNT FFFFF62EH

7.5 Timer Output Operations

The following table shows the operations and output levels of the TOBn0 and TOBm1 to TOBm3 pins.

Table 7-7. Timer Output Control in Each Mode

(a) TAB0 of V850E/IG4-H, TAB0 and TAB1 of V850E/IH4-H

Operation Mode	TOBn0 Pin	TOBn1 Pin	TOBn2 Pin	TOBn3 Pin
Interval timer mode	PWM output			
External event count mode	None			
External trigger pulse output mode	PWM output	External trigger pulse output	External trigger pulse output	External trigger pulse output
One-shot pulse output mode		One-shot pulse output	One-shot pulse output	One-shot pulse output
PWM output mode		PWM output	PWM output	PWM output
Free-running timer mode	PWM output (only when compare function is used)			
Pulse width measurement mode	None			

(b) TAB1 of V850E/IG4-H

Operation Mode	TOB10 Pin
Interval timer mode	PWM output
External event count mode	None
External trigger pulse output mode	PWM output
One-shot pulse output mode	
PWM output mode	
Free-running timer mode	PWM output (only when compare function is used)
Pulse width measurement mode	None

Remark n = 0, 1

Table 7-8. Truth Table of TOBna Pins Under Control of Timer Output Control Bits

TABnIOC0.TABnOLa Bit	TABnIOC0.TABnOEa Bit	TABnCTL0.TABnCE bit	Level of TOBna Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark V850E/IG4-H: a = 0 to 3 when n = 0
a = 0 when n = 1
V850E/IH4-H: n = 0, 1
a = 0 to 3

7.6 Operation

TABn can perform the following functions.

Table 7-9. TABn Specifications in Each Mode

(a) TAB0 of V850E/IG4-H, TAB0 and TAB1 of V850E/IH4-H

Operation	TABnCTL1.TABnEST Bit (Software Trigger Bit)	TRGBn Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode	Valid	Valid	Compare only	Batch write
One-shot pulse output mode	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switching enabled	Anytime write
Pulse width measurement mode	Invalid	Invalid	Capture only	Not applicable

(b) TAB1 of V850E/IG4-H

Operation	TAB1CTL1.TAB1EST Bit (Software Trigger Bit)	TRGB1 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode	Valid	Valid	Compare only	Batch write
One-shot pulse output mode	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Compare only	Anytime write
Pulse width measurement mode	None			

Remarks 1. TABn has a function to execute tuning with TAAn. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

2. n = 0, 1

(1) Counter basic operation

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

Remark n = 0, 1
a = 0 to 3

(a) Counter start operation

- In external event count mode
  When the TABnCTL0.TABnCE bit is set from 0 to 1, the 16-bit counter is set to 0000H.
  After that, it counts up to 0001H, 0002H, 0003H, … each time the valid edge of external event count input (EVTBn) is detected.
- In modes other than the above
  Starts counting from the default value FFFFH in all modes.
  It counts up from FFFFH to 0000H, 0001H, 0002H, 0003H, and so on.

(b) Clear operation

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and when its value is captured. The count operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTBnCCa interrupt signal is not generated.

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running timer mode or pulse width measurement mode (TAB0 and TAB1 (V850E/IH4-H) or TAB0 only (V850E/IG4-H)). If the counter overflows, the TABnOPT0.TABnOVF bit is set to 1 and an interrupt request signal (INTTBnOV) is generated. Note that the INTTBnOV signal is not generated under the following conditions.

- Immediately after a count operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured in the pulse width measurement mode and the counter counts up from FFFFH to 0000H

Caution After the overflow interrupt request signal (INTTBnOV) has been generated, be sure to check that the overflow flag (TABnOVF bit) is set to 1.

(d) Counter read operation during count operation

The value of the 16-bit counter of TABn can be read by using the TABnCNT register during the count operation.

When the TABnCTL0.TABnCE bit = 1, the value of the 16-bit counter can be read by reading the TABnCNT register. When the TABnCE bit = 0, the 16-bit counter is FFFFH and the TABnCNT register is 0000H.

(e) Interrupt operation

TABn generates the following five interrupt request signals.

- INTTBnCC0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TABmCCR0 register.
- INTTBnCC1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TABmCCR1 register.
- INTTBnCC2 interrupt: This signal functions as a match interrupt request signal of the CCR2 buffer register and as a capture interrupt request signal to the TABmCCR2 register.
- INTTBnCC3 interrupt: This signal functions as a match interrupt request signal of the CCR3 buffer register and as a capture interrupt request signal to the TABmCCR3 register.
- INTTBnOV interrupt: This signal functions as an overflow interrupt request signal.

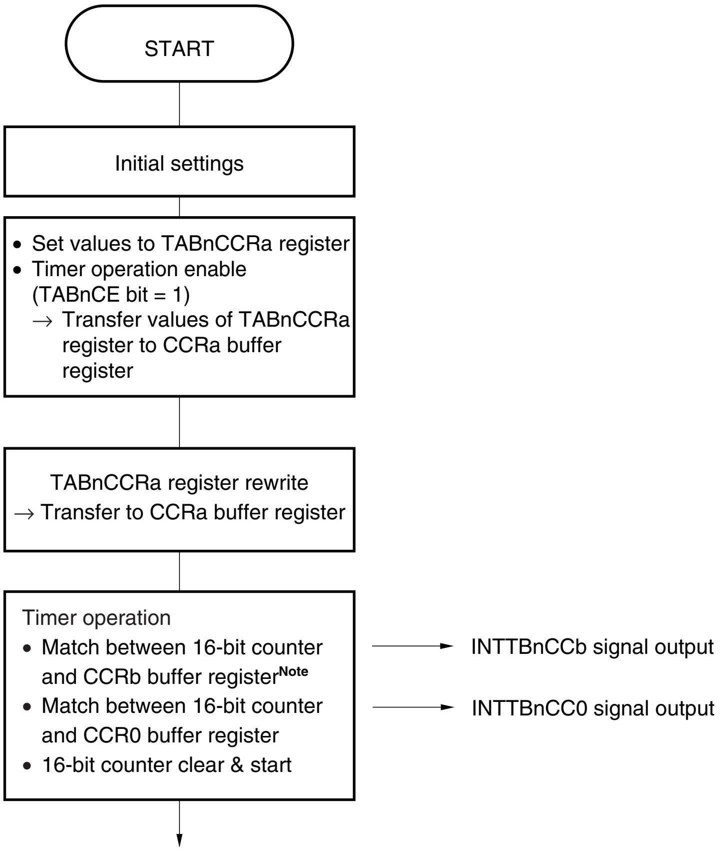
(2) Anytime write and batch write

The TABnCCR0 to TABnCCR3 registers can be rewritten in the TABn during timer operation (TABnCTL0.TABnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 to CCR3 buffer registers differs depending on the mode.

(a) Anytime write

In this mode, data is transferred at any time from the TABnCCR0 to TABnCCR3 registers to the CCR0 to CCR3 buffer registers during the timer operation.

Figure 7-3. Flowchart of Basic Operation for Anytime Write

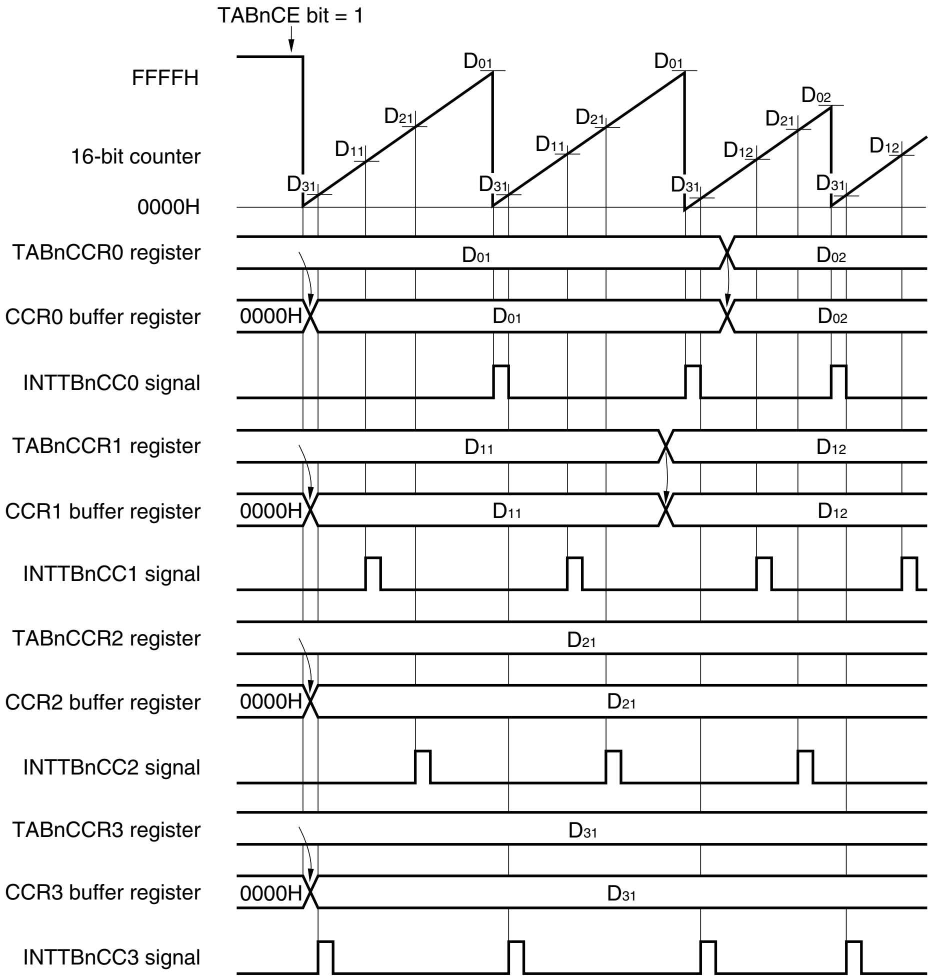

START
Initial settings
• Set values to TABnCCRa register
• Timer operation enable (TABnCE bit = 1)
→ Transfer values of TABnCCRa register to CCRa buffer register
TABnCCRa register rewrite
→ Transfer to CCRa buffer register
Timer operation
• Match between 16-bit counter and CCRb buffer registerNote
• Match between 16-bit counter and CCR0 buffer register
• 16-bit counter clear & start
INTTBnCCb signal output
INTTBnCC0 signal output

Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCRb buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Remarks 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0, 1
a = 0 to 3
b = 1 to 3

Figure 7-4. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Set values of TABnCCR0 register

D_{11}, D_{12}: Set values of TABnCCR1 register

D_{21}: Set value of TABnCCR2 register

D_{31}: Set value of TABnCCR3 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0, 1

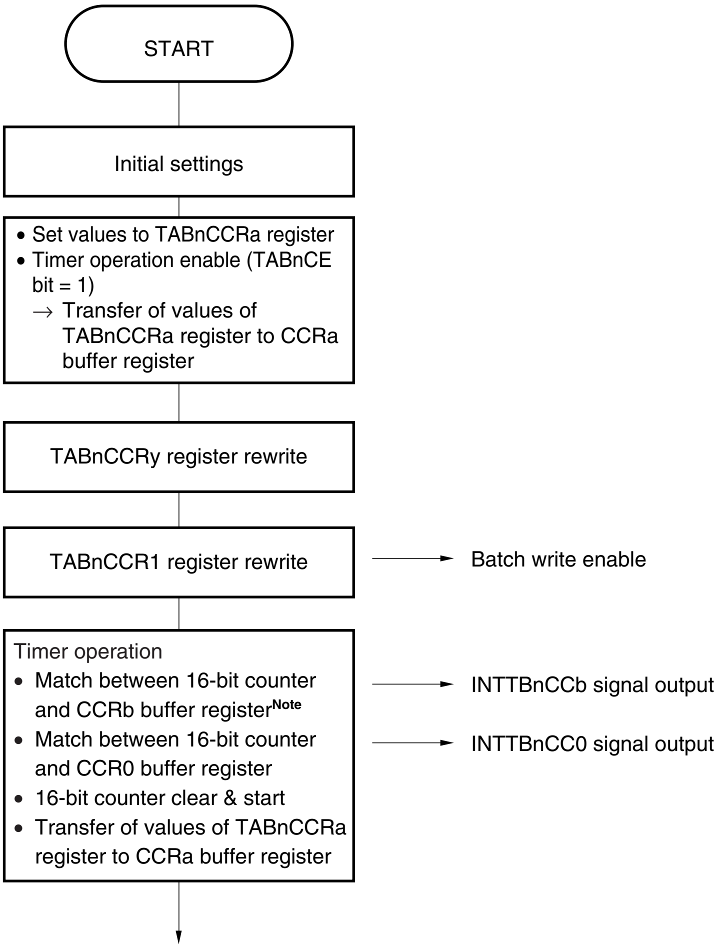
(b) Batch write

In this mode, data is transferred all at once from the TABnCCR0 to TABnCCR3 registers to the CCR0 to CCR3 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TABnCCR1 register.

Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TABnCCR1 register.

In order for the set value when the TABnCCR0 to TABnCCR3 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 to CCR3 buffer registers), it is necessary to rewrite TABnCCR0 and finally write to the TABnCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. The values of the TABnCCR0 to TABnCCR3 registers are transferred to the CCR0 to CCR3 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus, even when wishing only to rewrite the value of the TABnCCR0, TABnCCR2, or TABnCCR3 register, also write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

Figure 7-5. Flowchart of Basic Operation for Batch Write


START
Initial settings
• Set values to TABnCCRa register
• Timer operation enable (TABnCE bit = 1)
→ Transfer of values of TABnCCRa register to CCRa buffer register
TABnCCRy register rewrite
TABnCCR1 register rewrite
Batch write enable
Timer operation
• Match between 16-bit counter and CCRb buffer register[Note]
• Match between 16-bit counter and CCR0 buffer register
• 16-bit counter clear & start
• Transfer of values of TABnCCRa register to CCRa buffer register
INTTBnCCb signal output
INTTBnCC0 signal output

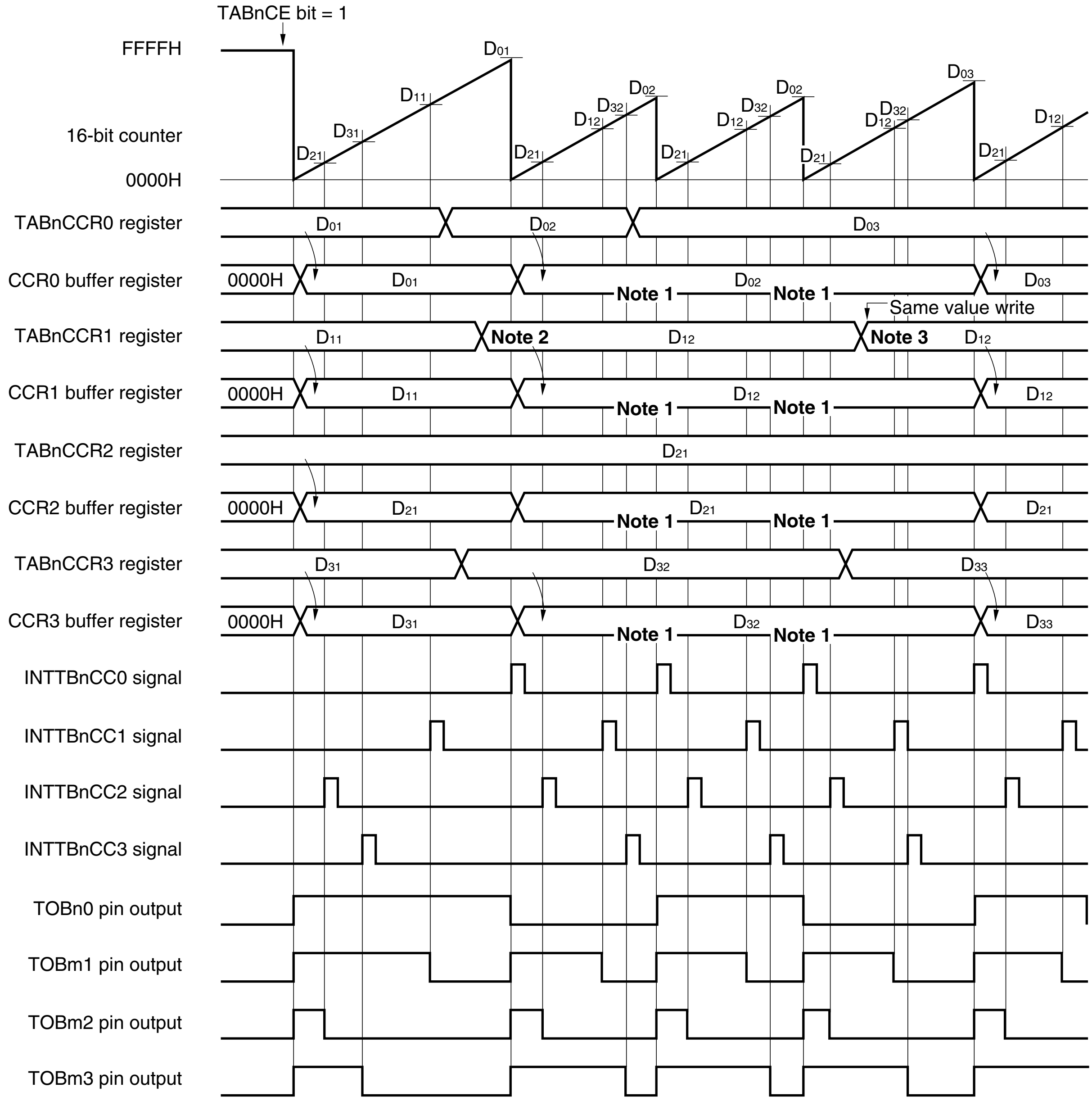
Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCRb buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Caution Writing to the TABnCCR1 register includes enabling of batch write. Thus, rewrite the TABnCCR1 register after rewriting the TABnCCR0, TABnCCR2, and TABnCCR3 registers.

Remarks 1. The above flowchart illustrates an example of the operation in the PWM output mode.

2. a = 0 to 3
b = 1 to 3
y = 0, 2, 3

Figure 7-6. Timing of Batch Write



Notes 1. Because the TABnCCR1 register was not rewritten, D02 is not transferred.

2. Because TABnCCR1 register has been written (D12), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit timer and the value of the TABnCCR0 register (D01).

3. Because TABnCCR1 register has been written (D12), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit timer and the value of the TABnCCR0 register (D12).

Remarks 1. D01, D02, D03: Set values of TABnCCR0 register
D11, D12: Set values of TABnCCR1 register
D21: Set value of TABnCCR2 register
D31, D32, D33: Set values of TABnCCR3 register

2. The above timing chart illustrates the operation in the PWM output mode as an example.

3. V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

7.6.1 Interval timer mode (TABnMD2 to TABnMD0 bits = 000)

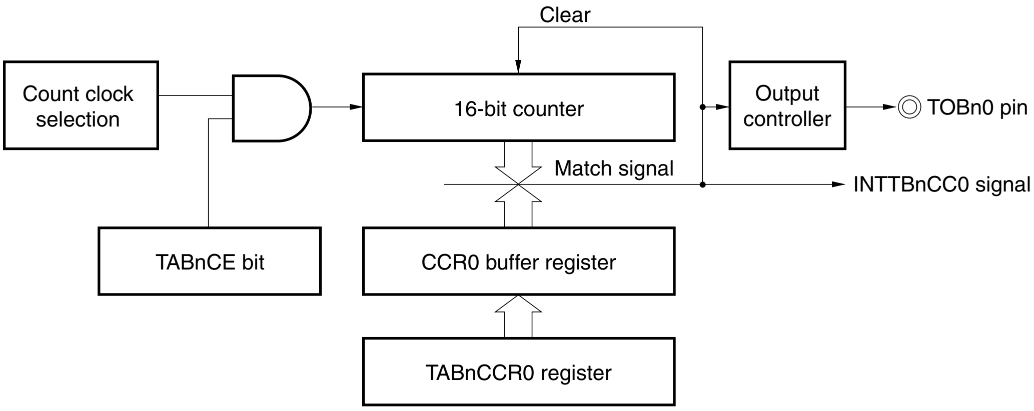
In the interval timer mode, an interrupt request signal (INTTBnCC0) is generated at the interval set by the TABnCCR0 register if the TABnCTL0.TABnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOBn0 pin.

The TABnCCR1 to TABnCCR3 registers are not used in the interval timer mode. However, the set value of the TABnCCR1 to TABnCCR3 registers is transferred to the CCR1 to CCR3 buffer registers and, when the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, compare match interrupt request signals (INTTBnCC1 to INTTBnCC3) are generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTBmCC1 to INTTBmCC3 signals are generated, can be output from the TOBm1 to TOBm3 pins.

The value of the TABnCCR1 to TABnCCR3 registers can be rewritten even while the timer is operating.

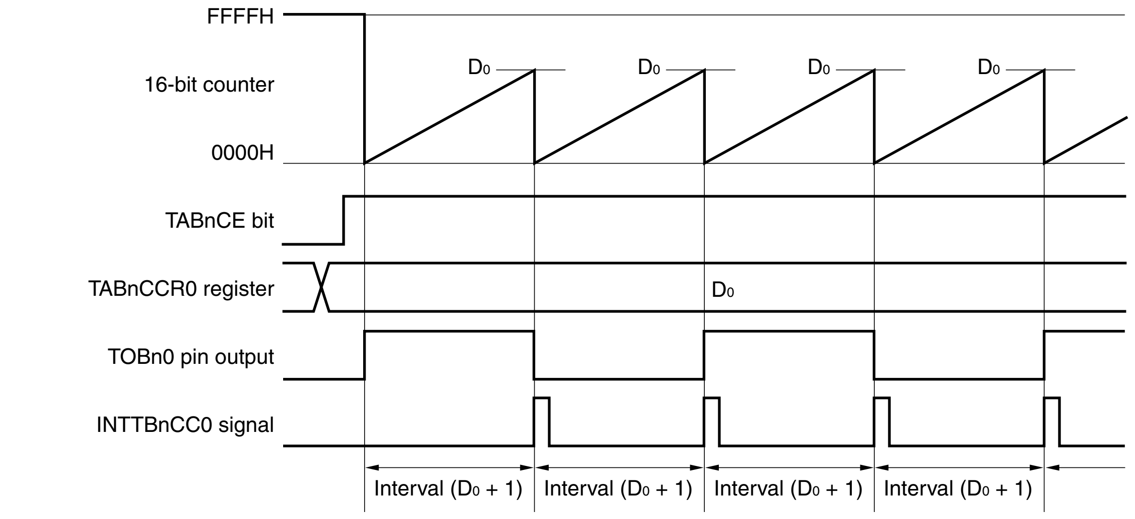
Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

Figure 7-7. Interval Timer Configuration



Remark n = 0, 1

Figure 7-8. Basic Timing of Operation in Interval Timer Mode



Remark n = 0, 1

When the TABnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TOBn0 pin is inverted. Additionally, the set value of the TABnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TOBn0 pin is inverted, and a compare match interrupt request signal (INTTBnCC0) is generated.

The interval can be calculated by the following expression.

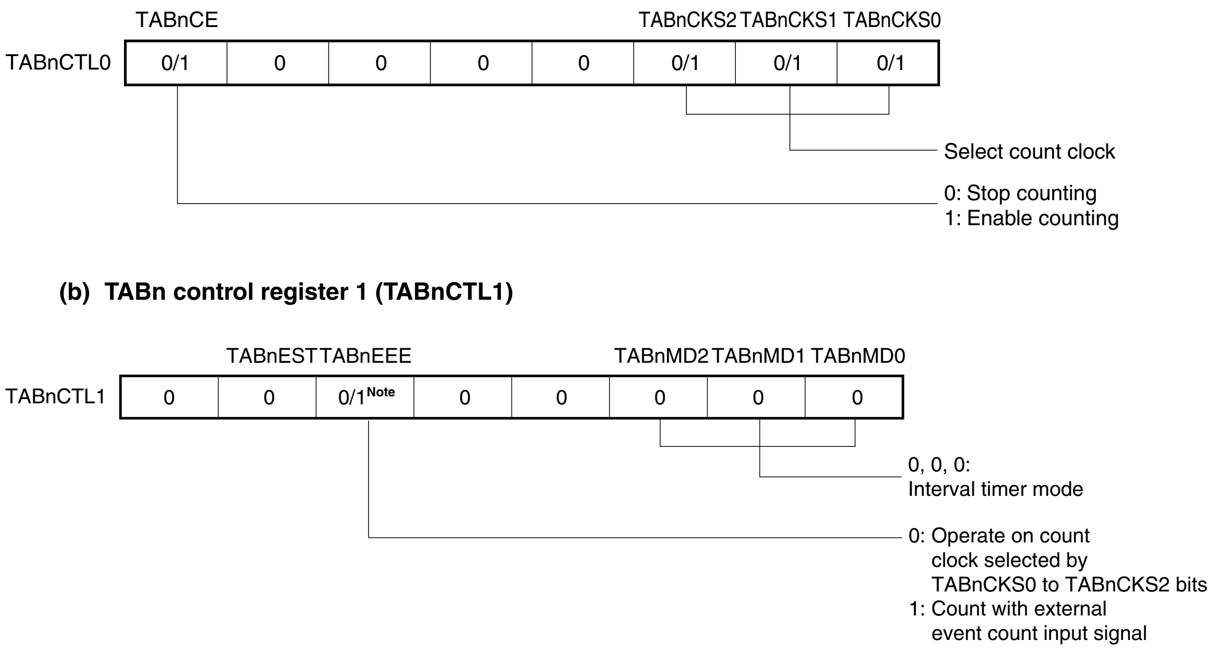
Interval = (Set value of TABnCCR0 register + 1) × Count clock cycle

Remark n = 0, 1

Figure 7-9. Register Setting for Interval Timer Mode Operation (1/3)

(a) TABn control register 0 (TABnCTL0)

(b) TABn control register 1 (TABnCTL1)



Note The TABnEEE bit can be set to 1 only when timer output (TOBn0 andTOBmb) is used. However, set the TABnCCR0 to TABnCCR3 registers to the same value.

Figure 7-9. Register Setting for Interval Timer Mode Operation (2/3)

(c) TABn I/O control register 0 (TABnIOC0)

	TABmOL3	TABmOE3	TABmOL2	TABmOE2	TABmOL1	TABmOE1	TABmOL0	TABmOE0
TABnIOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

0: Disable TOBn0 pin output
1: Enable TOBn0 pin output

Setting of TOBn0 pin output level before count operation
0: Low level
1: High level

0: Disable TOBm1 pin output
1: Enable TOBm1 pin output

Setting of TOBm1 pin output level before count operation
0: Low level
1: High level

0: Disable TOBm2 pin output
1: Enable TOBm2 pin output

Setting of TOBm2 pin output level before count operation
0: Low level
1: High level

0: Disable TOBm3 pin output
1: Enable TOBm3 pin output

Setting of TOBm3 pin output level before count operation
0: Low level
1: High level

(d) TABn I/O control register 2 (TABnIOC2)

					TABnEES1	TABnEES0	TABnETS1	TABnETS0
TABnIOC2	0	0	0	0	0/1Note	0/1Note	0	0

Select valid edge of external event count input (EVTBn pin).

Note The TABnEES1 and TABnEES0 bits can be set only when timer output (TOBn0, TOBm1 to TOBm3) is used. However, set the TABnCCR0 to TABnCCR3 registers to the same value.

(e) TABn counter read buffer register (TABnCNT)

By reading the TABnCNT register, the count value of the 16-bit counter can be read.

(f) TABn capture/compare register 0 (TABnCCR0)

If the TABnCCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ Count clock cycle

Figure 7-9. Register Setting for Interval Timer Mode Operation (3/3)

(g) TABn capture/compare registers 1 to 3 (TABnCCR1 to TABnCCR3)

The TABnCCR1 to TABnCCR3 registers are not used in the interval timer mode. However, the set values of the TABnCCR1 to TABnCCR3 registers are transferred to the CCR1 to CCR3 buffer registers. When the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, the TOBm1 to TOBm3 pin outputs are inverted and a compare match interrupt request signal (INTTBmCC1 to INTTBmCC3) is generated.

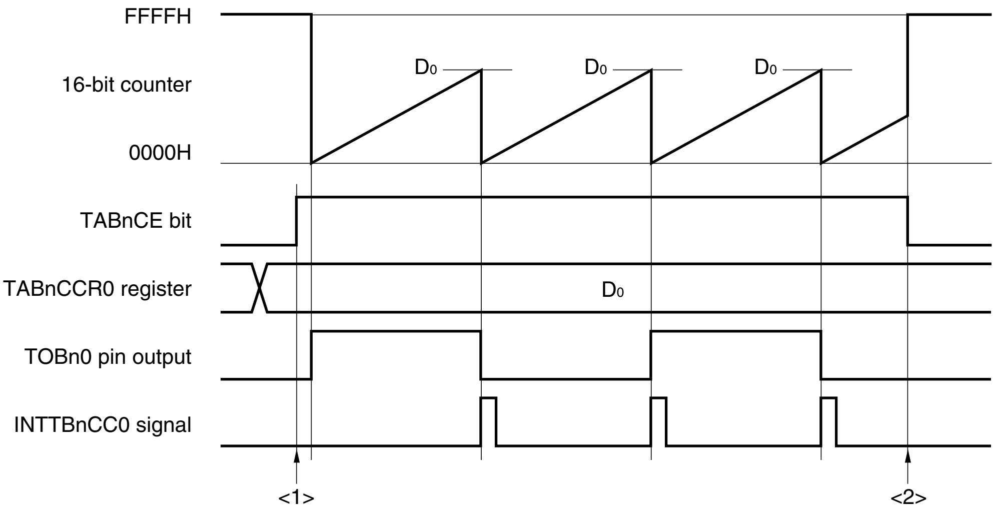
When the TABnCCR1 to TABnCCR3 registers are not used, it is recommended to set their values to FFFFH. Also mask the registers by the interrupt mask flags (TABnCCIC1.TABnCCMK1 to TABnCCIC3.TABnCCMK3).

Remarks 1. TABm I/O control register 1 (TABmIOC1) and TABn option register 0 (TABnOPT0) are not used in the interval timer mode.

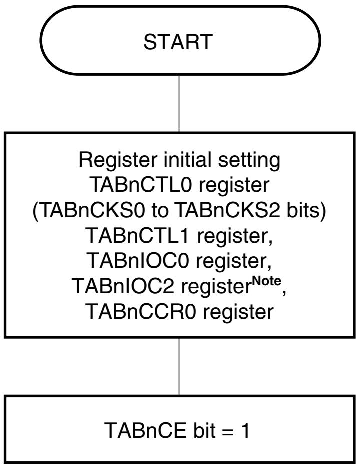
2. V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

(1) Interval timer mode operation flow

Figure 7-10. Software Processing Flow in Interval Timer Mode (1/2)



<1> Count operation start flow



Initial setting of these registers is performed before setting the TABnCE bit to 1.

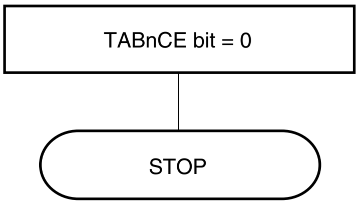
The TABnCKS0 to TABnCKS2 bits can be set at the same time as when counting starts (TABnCE bit = 1).

Note The TABnEES1 and TABnEES0 bits can be set only when timer output (TOBn0, TOBm1 to TOBm3) is used. However, set the TABnCCR0 to TABnCCR3 registers to the same value.

Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

Figure 7-10. Software Processing Flow in Interval Timer Mode (2/2)

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TABnCE bit to 0. The output level of the TOBn0 pin is as specified by the TABnIOC0 register.

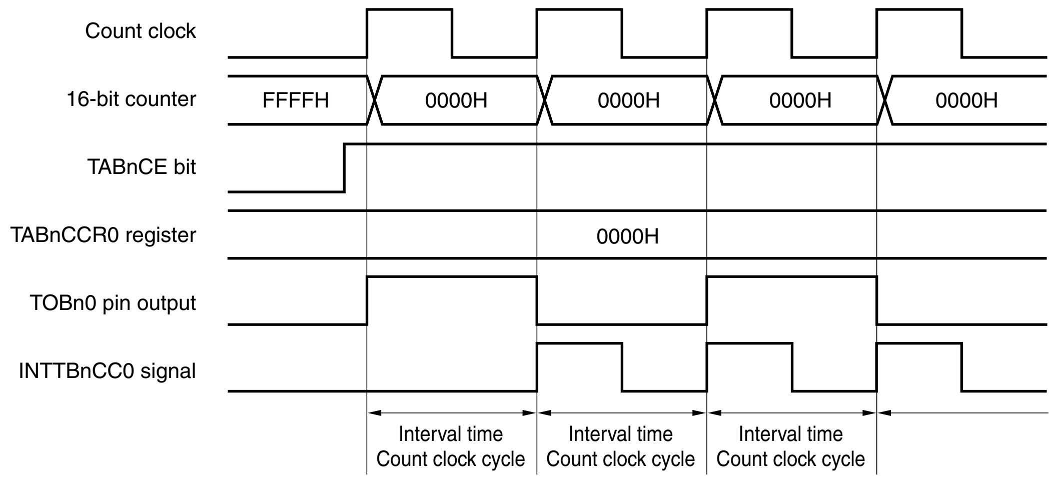
Remark n = 0, 1

(2) Interval timer mode operation timing

(a) Operation if TABnCCR0 register is set to 0000H

If the TABnCCR0 register is set to 0000H, the INTTBnCC0 signal is generated at each count clock, and the output of the TOBn0 pin is inverted.

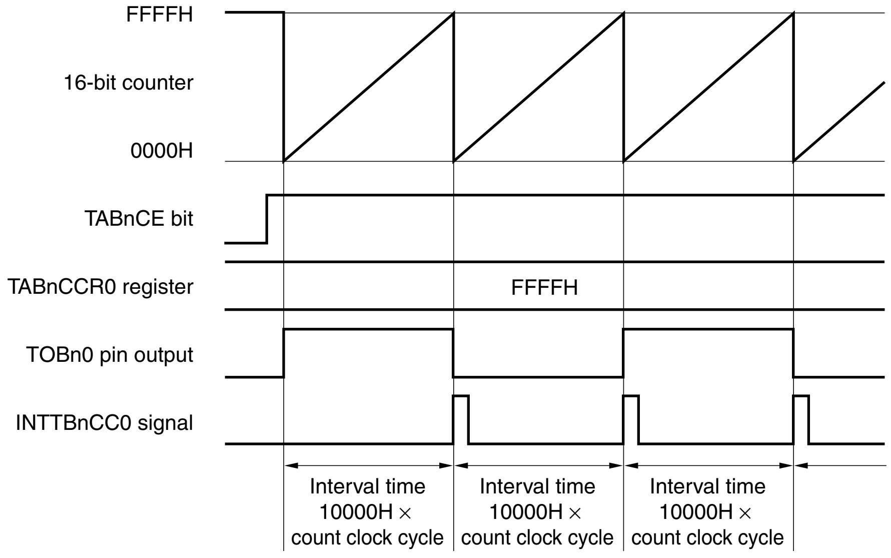
The value of the 16-bit counter is always 0000H.



Remark n = 0, 1

(b) Operation if TABnCCR0 register is set to FFFFH

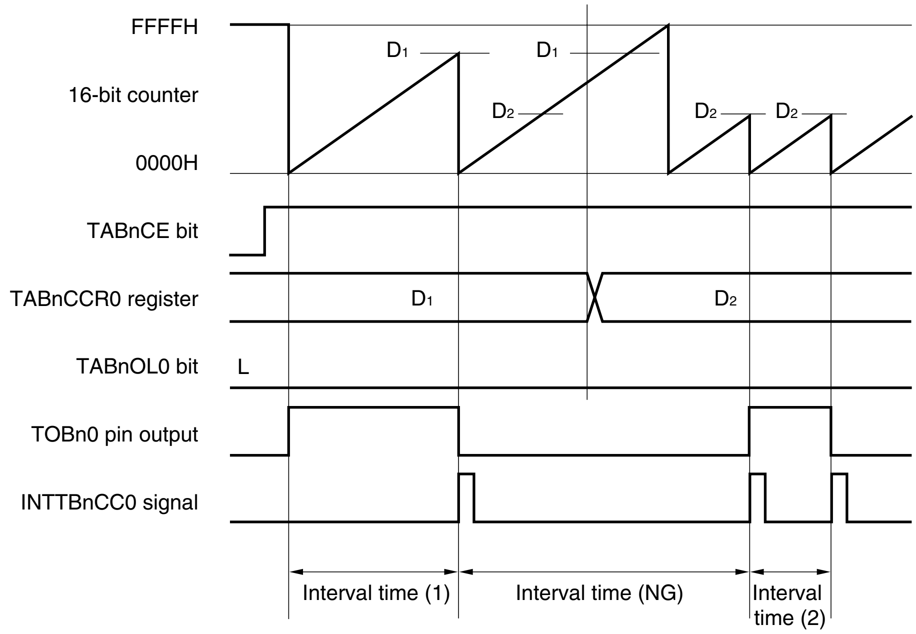
If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTBnCC0 signal is generated and the output of the TOBn0 pin is inverted. At this time, an overflow interrupt request signal (INTTBnOV) is not generated, nor is the overflow flag (TABnOPT0.TABnOVF bit) set to 1.



Remark n = 0, 1

(c) Notes on rewriting TABnCCR0 register

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



Remarks 1. Interval time (1): (D_1 + 1) × Count clock cycle

Interval time (NG): (10000H + D_2 + 1) × Count clock cycle

Interval time (2): (D_2 + 1) × Count clock cycle

2. n = 0, 1

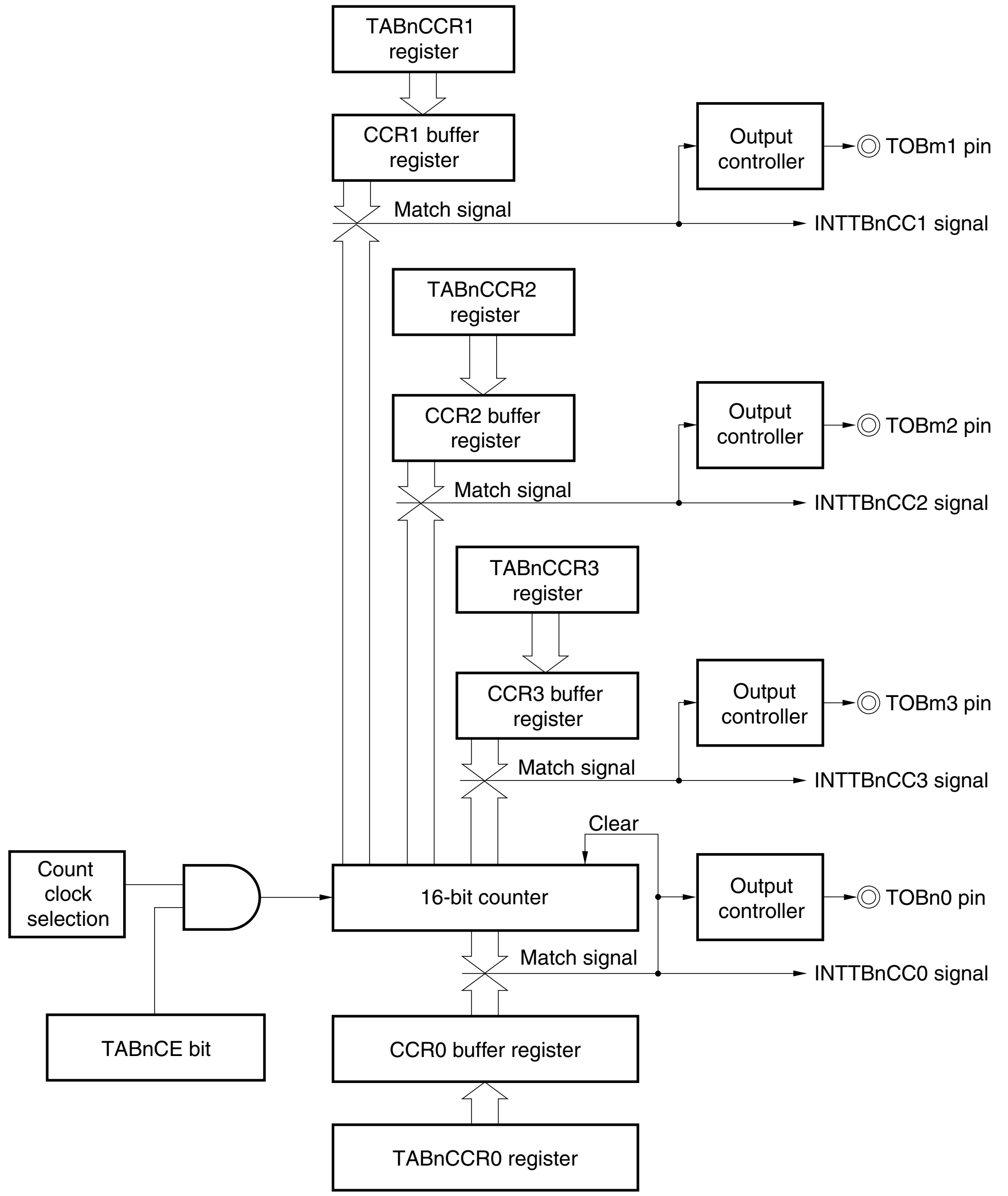
If the value of the TABnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTBnCC0 signal is generated and the output of the TOBn0 pin is inverted.

Therefore, the INTTBnCC0 signal may not be generated at the interval time "(D_1 + 1) × Count clock cycle" or "(D_2 + 1) × Count clock cycle" originally expected, but may be generated at an interval of "(10000H + D_2 + 1) × Count clock period".

(d) Operation of TABnCCR1 to TABnCCR3 registers

Figure 7-11. Configuration of TABnCCR1 to TABnCCR3 Registers



Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

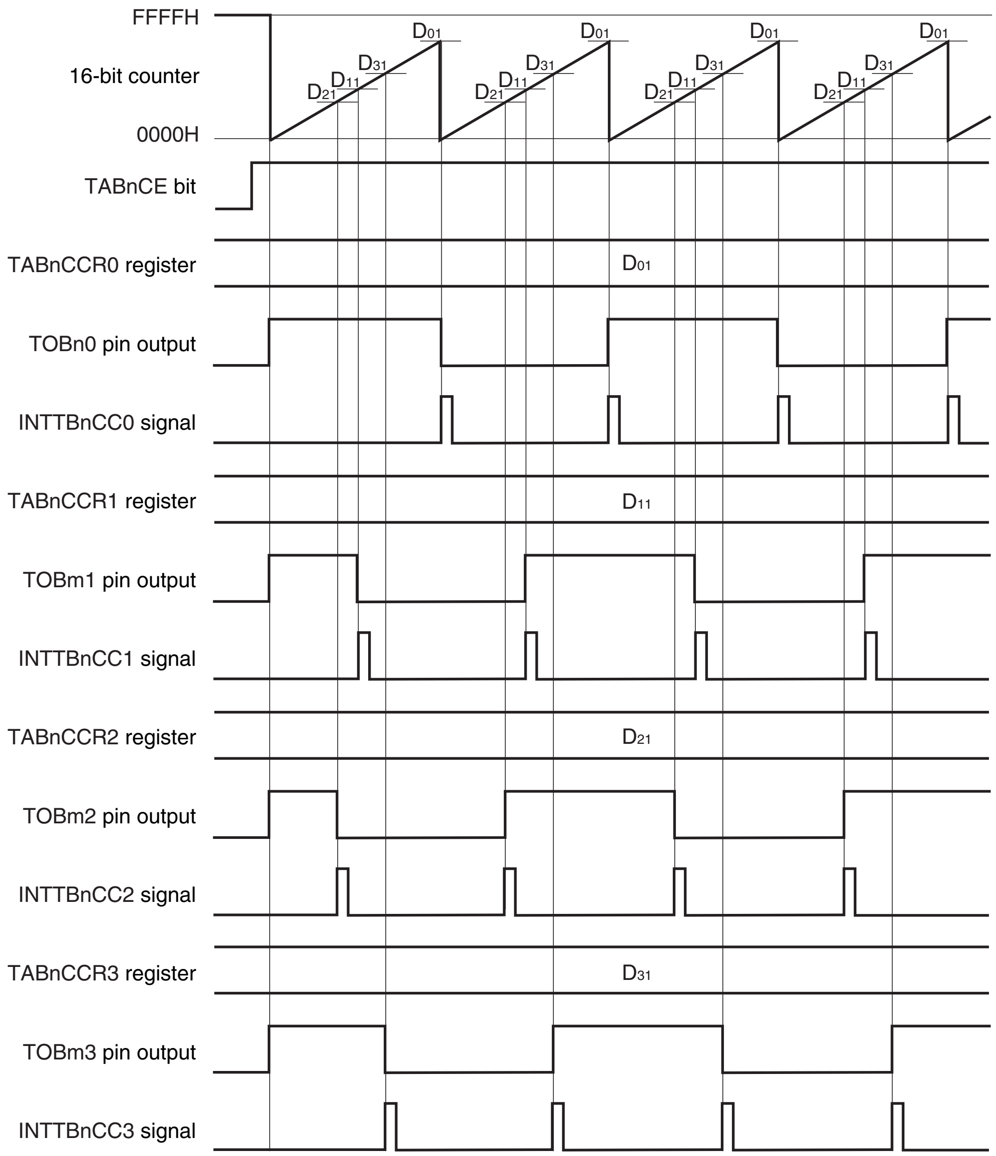
When the TABnCCRb register is set to the same value as the TABnCCR0 register, the INTTBnCCb signal is generated at the same timing as the INTTBnCC0 signal and the TOBmb pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOBmb pin.

The following shows the operation when the TABnCCRb register is set to other than the value set in the TABnCCR0 register.

If the set value of the TABnCCRb register is less than the set value of the TABnCCR0 register, the INTTBnCCb signal is generated once per cycle. At the same time, the output of the TOBmb pin is inverted.

The TOBmb pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

Figure 7-12. Timing Chart When $D_{01} \geq D_{b1}$

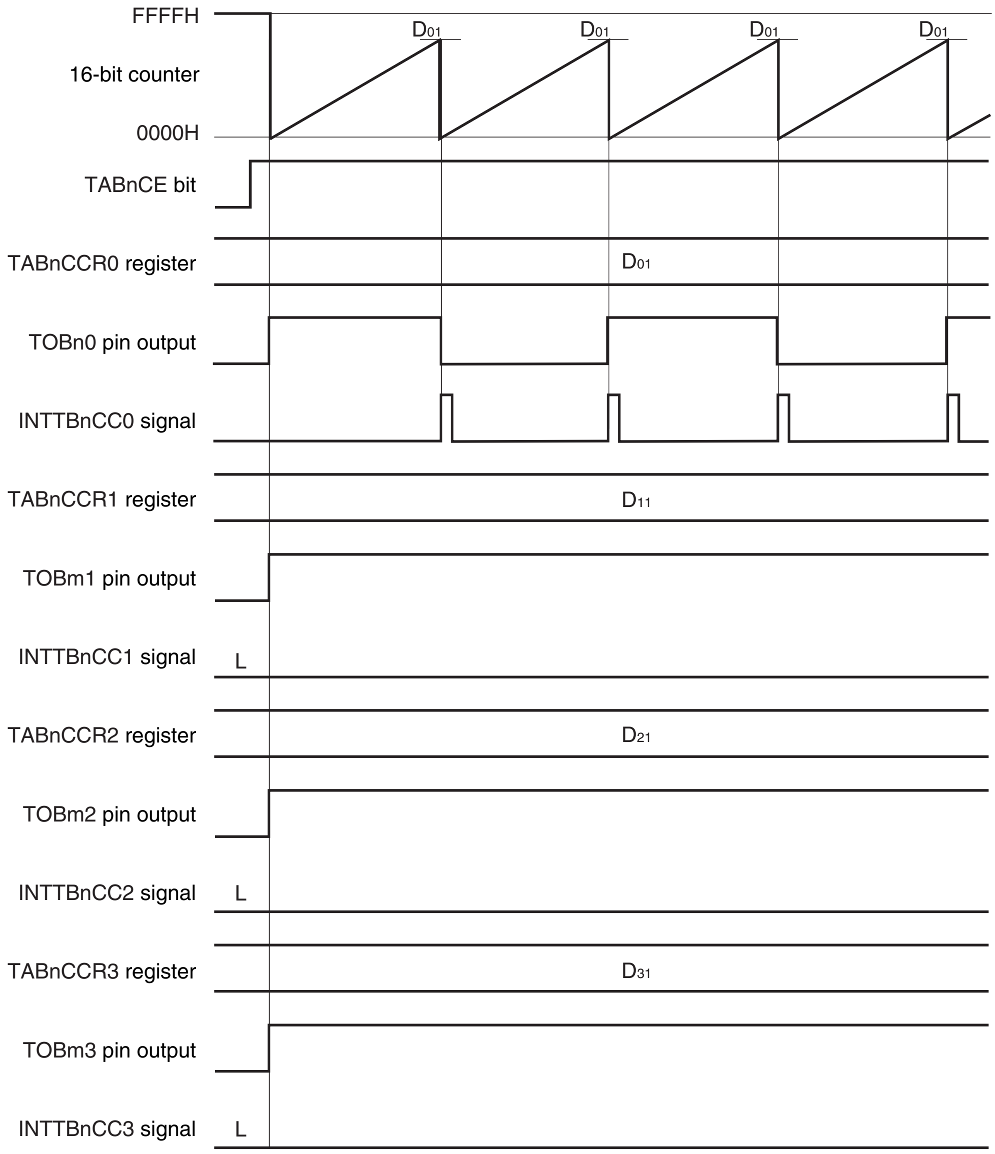


Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

If the set value of the TABnCCRb register is greater than the set value of the TABnCCR0 register, the count value of the 16-bit counter does not match the value of the TABnCCRb register. Consequently, the INTTBnCCb signal is not generated, nor is the output of the TOBmb pin changed.

When the TABnCCRb register is not used, it is recommended to set its value to FFFFH.

Figure 7-13. Timing Chart When $D_{01} < D_{b1}$



Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3

V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

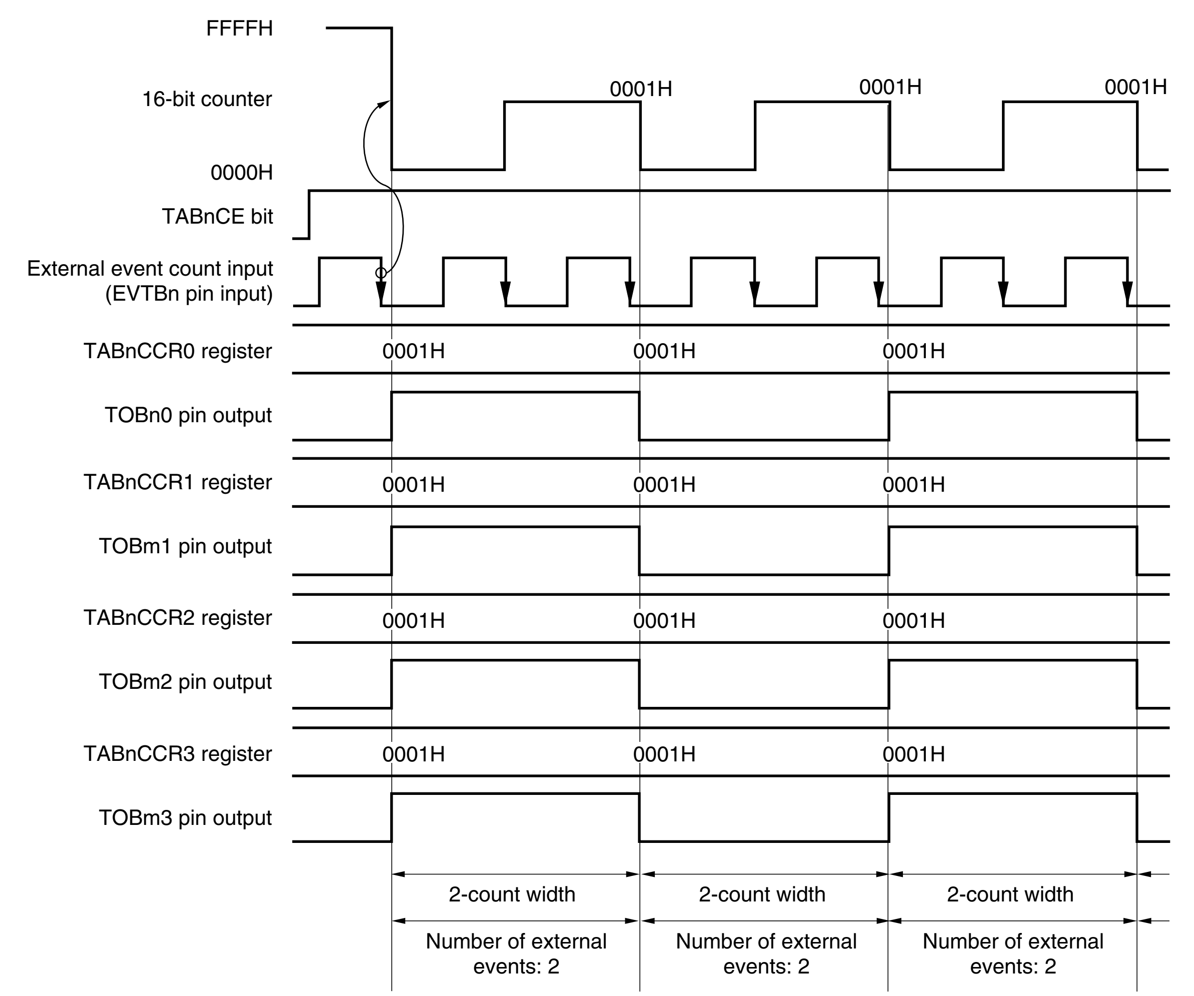
(3) Operation by external event count input (EVTBn)

(a) Operation

To count the 16-bit counter at the valid edge of the external event count input (EVTBn) in the interval timer mode, the 16-bit counter is cleared from FFFFH to 0000H by the valid edge of the external event count input after the TABnCE bit is set from 0 to 1.

When 0001H is set to both the TABnCCR0 and TABnCCRb registers, the output of the TOBn0 and TOBmb pins is inverted each time the 16-bit counter counts twice (b = 1 to 3).

The TABnCTL1.TABnEEE bit can be set to 1 in the interval timer mode only when the timer output (TOBn0, TOBmb) is used with the external event count input.



Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

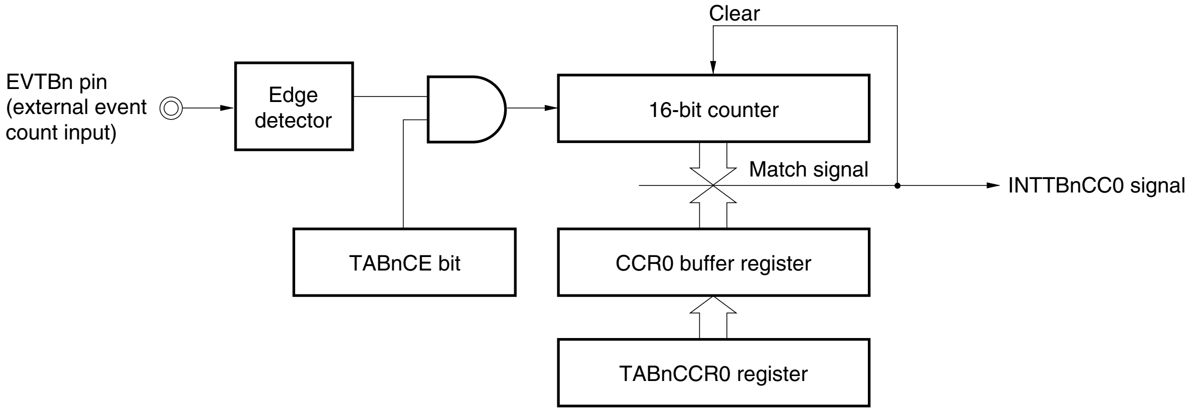
7.6.2 External event count mode (TABnMD2 to TABnMD0 bits = 001)

In the external event count mode, the valid edge of the external event count input (EVTBn) is counted when the TABnCTL0.TABnCE bit is set to 1, and an interrupt request signal (INTTBnCC0) is generated each time the specified number of edges set by the TABnCCR0 register have been counted. The TOBn0 and TOBm1 to TOBm3 pins cannot be used. When using the TOBn0 and TOBm1 to TOBm3 pins for external event count input, set the TABnCTL1.TABnEEE bit to 1 in the interval timer mode (see **7.6.1 (3) Operation by external event count input (EVTBn)**).

The TABnCCR1 to TABnCCR3 registers are not used in the external event count mode.

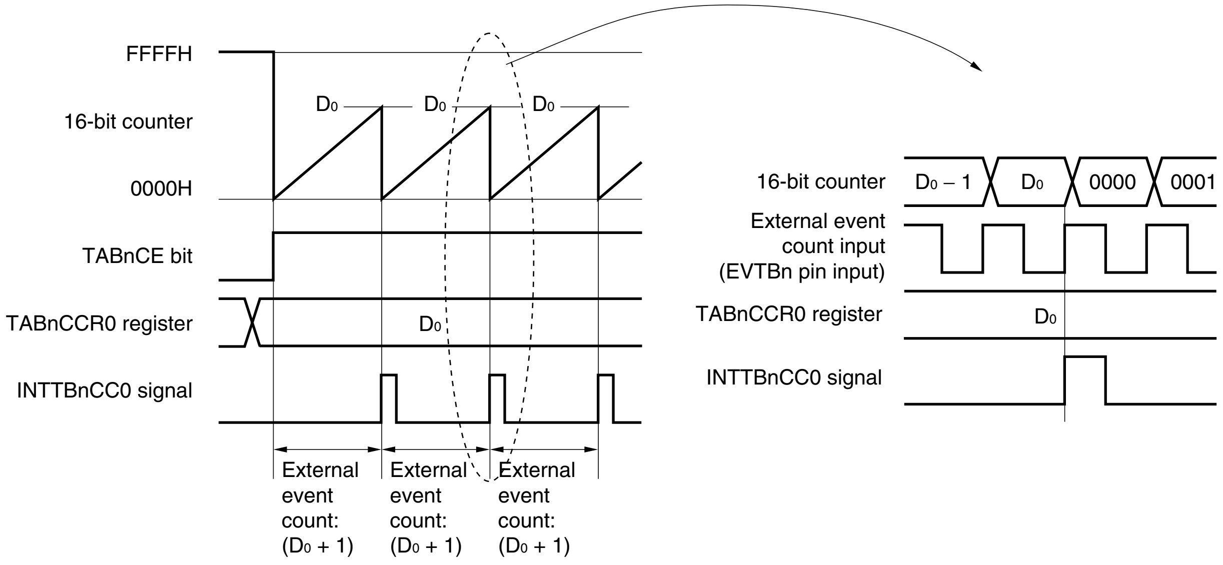
Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

Figure 7-14. Configuration in External Event Count Mode



Remark n = 0, 1

Figure 7-15. Basic Timing in External Event Count Mode



Remarks 1. This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

2. n = 0, 1

When the TABnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TABnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTBnCC0) is generated.

The INTTBnCC0 signal is generated each time the valid edge of the external event count has been detected "value set to TABnCCR0 register + 1" times.

Figure 7-16. Register Setting for Operation in External Event Count Mode (1/2)

(a) TABn control register 0 (TABnCTL0)

	TABnCE					TABnCKS2	TABnCKS1	TABnCKS0
TABnCTL0	0/1	0	0	0	0	0	0	0

0: Stop counting
1: Enable counting

(b) TABn control register 1 (TABnCTL1)

		TABnEST	TABnEEE			TABnMD2	TABnMD1	TABnMD0
TABnCTL1	0	0	0	0	0	0	0	1

0, 0, 1:
External event count mode

(c) TABn I/O control register 2 (TABnIOC2)

					TABnEES1	TABnEES0	TABnETS1	TABnETS0
TABnIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge
of external event
count input (EVTBn pin)

(d) TABn counter read buffer register (TABnCNT)

The count value of the 16-bit counter can be read by reading the TABnCNT register.

(e) TABn capture/compare register 0 (TABnCCR0)

If the TABnCCR0 register is set to D_0, the compare match interrupt request signal (INTTBnCC0) is generated when the number of external events has reached (D_0 + 1).

Figure 7-16. Register Setting for Operation in External Event Count Mode (2/2)

(f) TABn capture/compare registers 1 to 3 (TABnCCR1 to TABnCCR3)

The TABnCCR1 to TABnCCR3 registers are not used in the external event count mode. However, the set value of the TABnCCR1 to TABnCCR3 registers are transferred to the CCR1 to CCR3 buffer registers. When the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, compare match interrupt request signals (INTTBnCC1 to INTTBnCC3) are generated.
When the TABnCCR1 to TABnCCR3 registers are not used, it is recommended to set their values to FFFFH. Also mask the registers by the interrupt mask flags (TABnCCIC1.TABnCCMK1 to TABnCCIC3.TABnCCMK3).

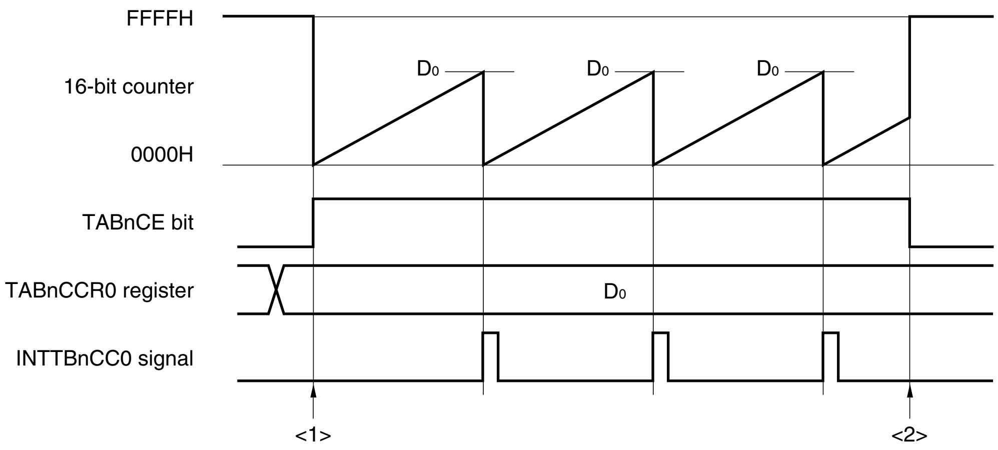
Caution Set the TABnIOC0 register to 00H.

Remarks 1. TABm I/O control register 1 (TABmIOC1) and TABn option register 0 (TABnOPT0) are not used in the external event count mode.

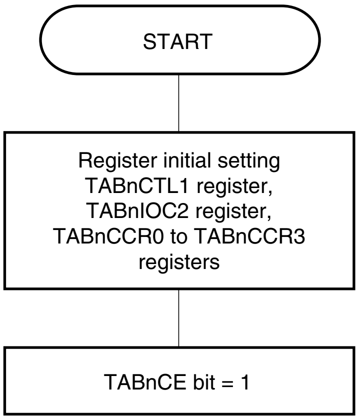
2. V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

(1) External event count mode operation flow

Figure 7-17. Software Processing Flow in External Event Count Mode

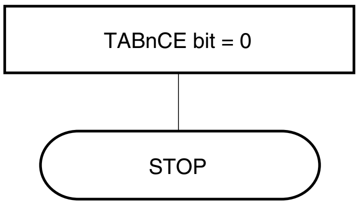


<1> Count operation start flow



Initial setting of these registers is performed before setting the TABnCE bit to 1.

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TABnCE bit to 0.

Remark n = 0, 1

(2) Operation timing in external event count mode

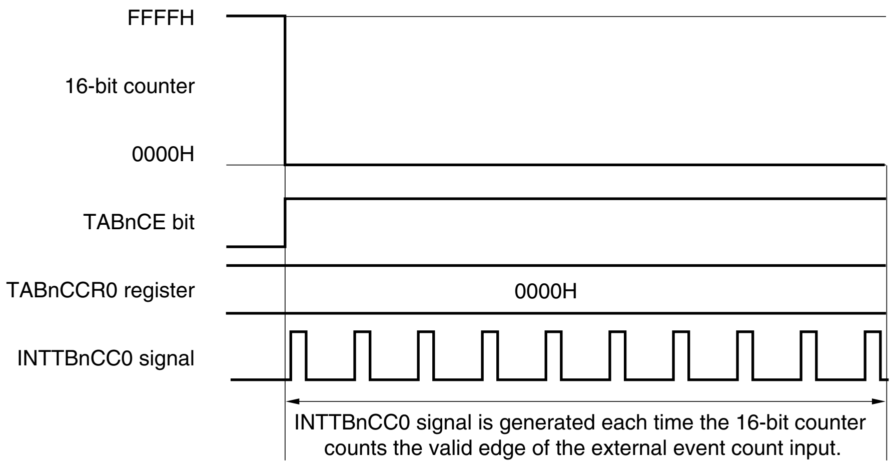
Caution In the external event count mode, use of the timer output (TOBn0, TOBm1 to TOBm3) is disabled. If using timer output (TOBn0, TOBm1 to TOBm3) with external event count input (EVTBn), set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TABnCTL1.TABnEEE bit = 1) (see 7.6.1 (3) Operation by external event count input (EVTBn)).

Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

(a) Operation if TABnCCR0 register is set to 0000H

When the TABnCCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates an INTTBnCC0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

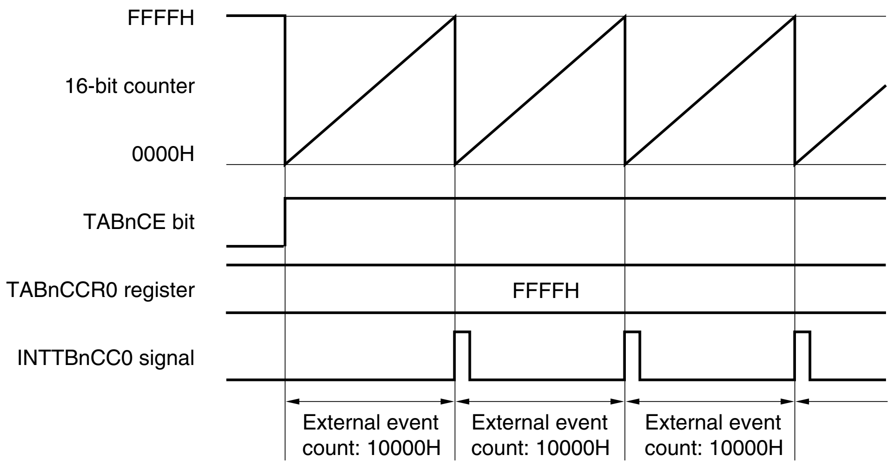
The value of the 16-bit counter is always 0000H.


FFFFH
16-bit counter
0000H
TABnCE bit
TABnCCR0 register
0000H
INTTBnCC0 signal
INTTBnCC0 signal is generated each time the 16-bit counter counts the valid edge of the external event count input.

Remark n = 0, 1

(b) Operation if TABnCCR0 register is set to FFFFH

If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTBnCC0 signal is generated. At this time, the TABnOPT0.TABnOVF bit is not set.

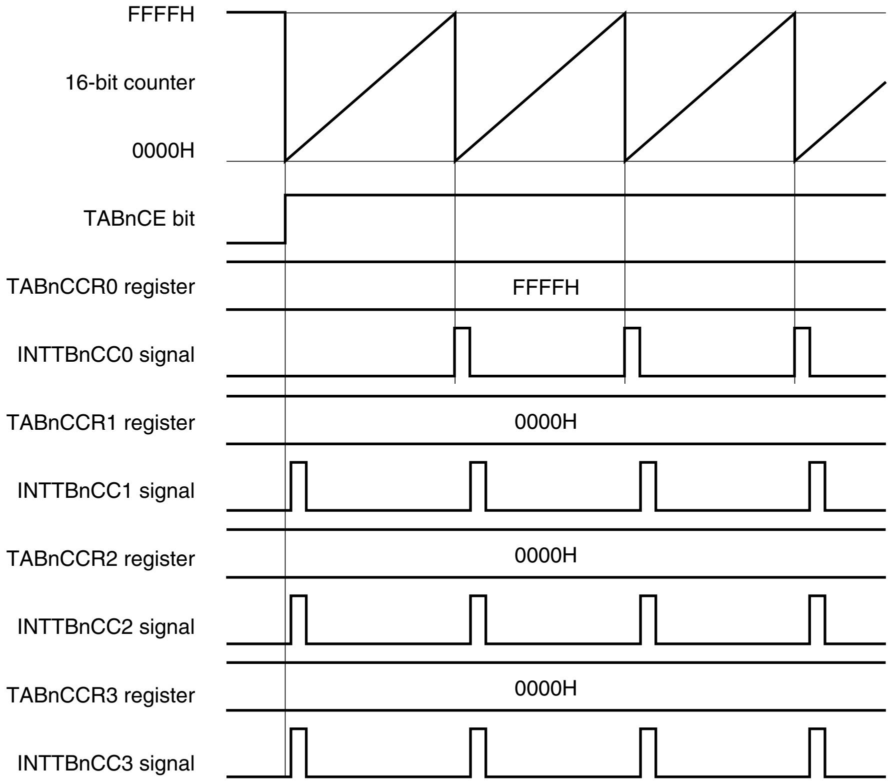

FFFFH
16-bit counter
0000H
TABnCE bit
TABnCCR0 register
FFFFH
INTTBnCC0 signal
External event count: 10000H
External event count: 10000H
External event count: 10000H

Remark n = 0, 1

(c) Operation with TABnCCR0 set to FFFFH and TABnCCRb register to 0000H

When the TABnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTBnCC0 signal is generated. At this time, the TABnOPT0.TABnOVF bit is not set.

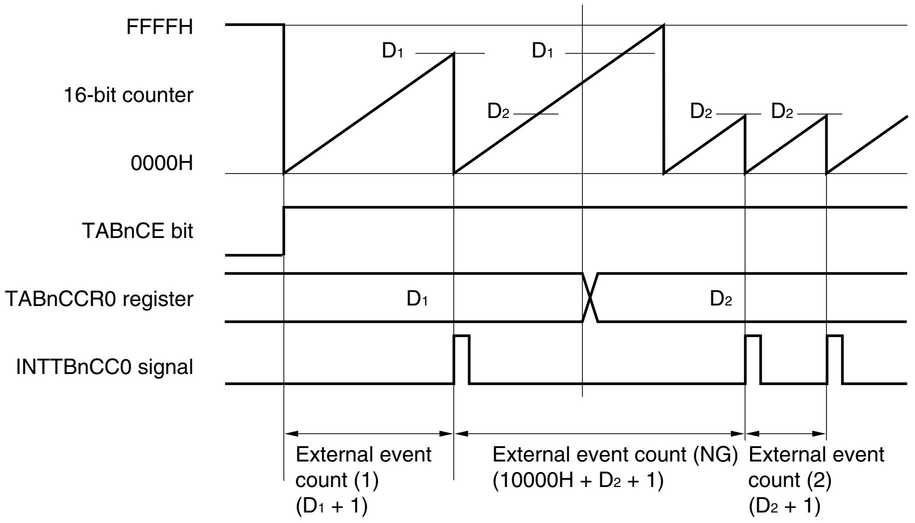
If the TABnCCRb register is set to 0000H, the INTTBnCCb signal is generated when the 16-bit counter is cleared to 0000H.



Remark n = 0, 1
b = 1 to 3

(d) Notes on rewriting the TABnCCR0 register

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.


FFFFH
16-bit counter
0000H
TABnCE bit
TABnCCR0 register
INTTBnCC0 signal
D1
D1
D2
D2
D2
D1
D2
External event count (1) (D1 + 1)
External event count (NG) (10000H + D2 + 1)
External event count (2) (D2 + 1)

Remark n = 0, 1

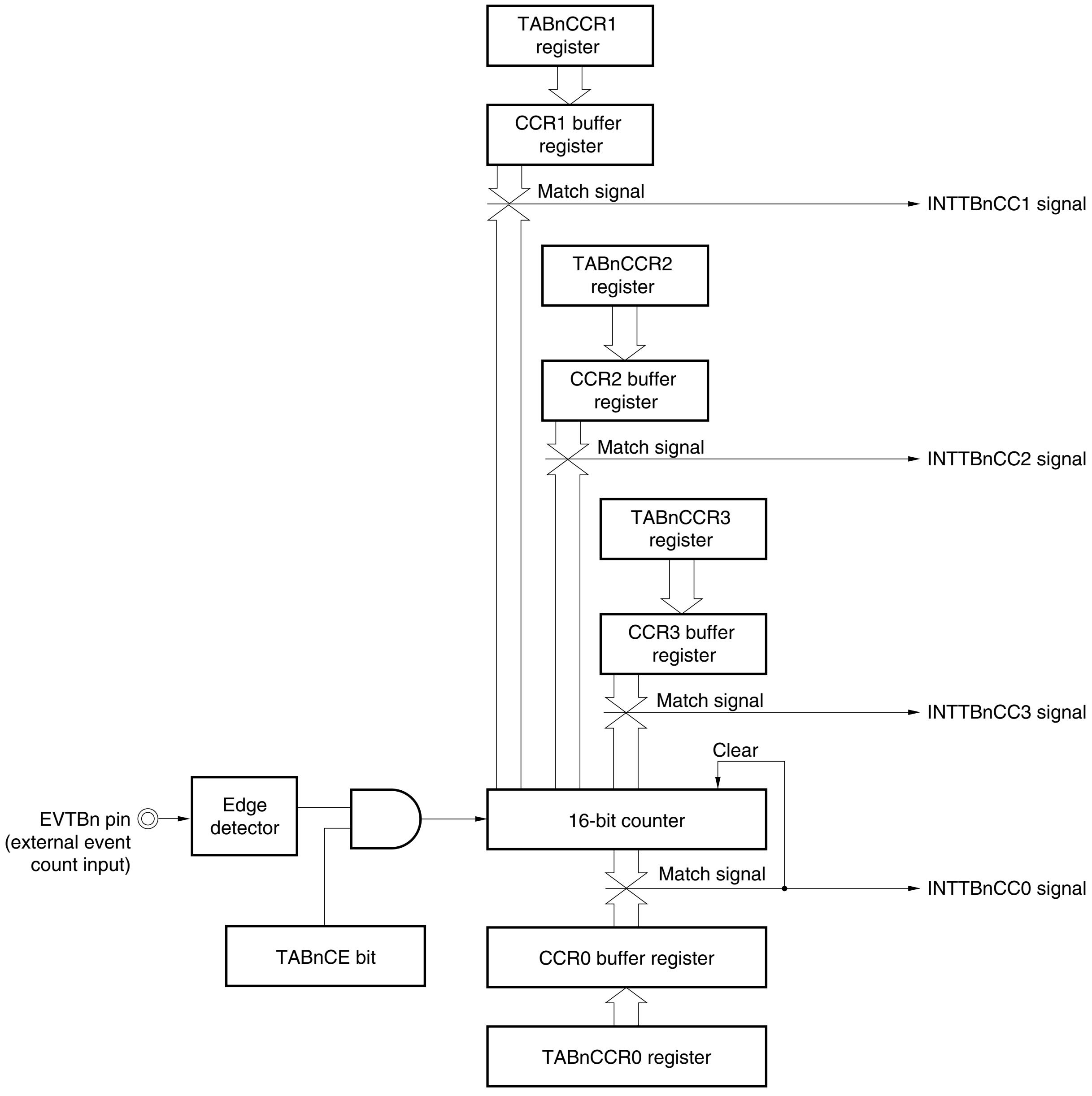
If the value of the TABnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTBnCC0 signal is generated.

Therefore, the INTTBnCC0 signal may not be generated at the valid edge count of "(D_1 + 1) times" or "(D_2 + 1) times" originally expected, but may be generated at the valid edge count of "(10000H + D_2 + 1) times".

(e) **Operation of TABnCCR1 to TABnCCR3 registers**

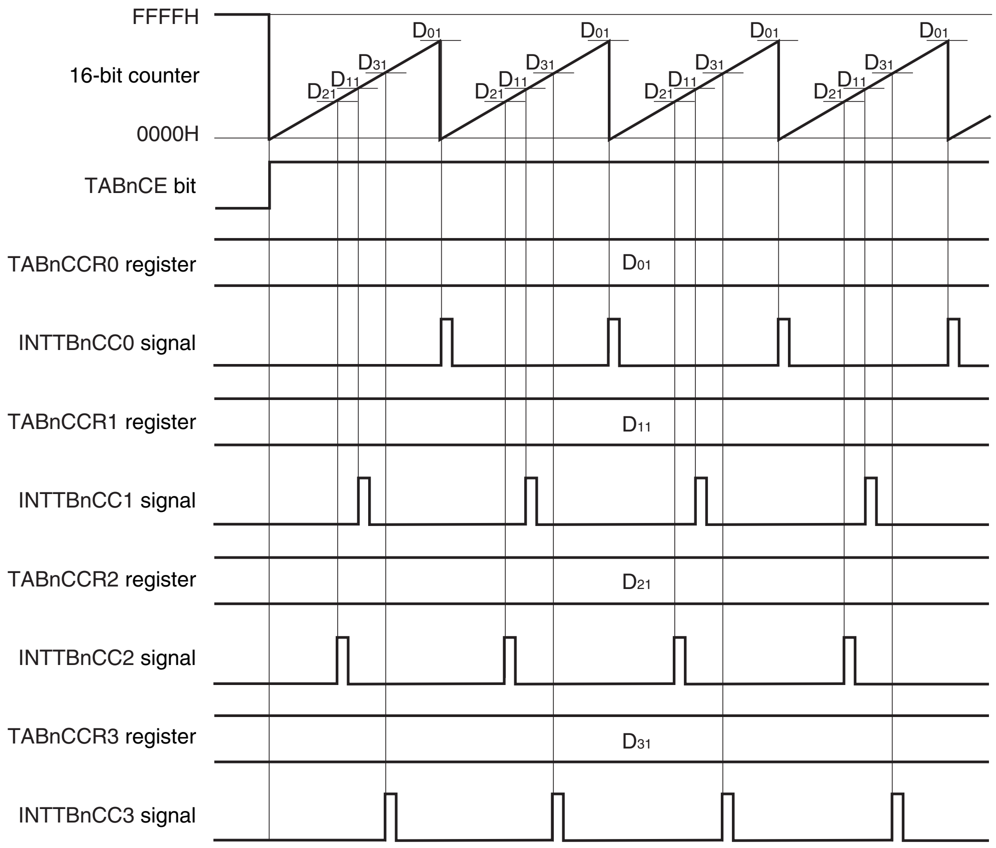
Figure 7-18. Configuration of TABnCCR1 to TABnCCR3 Registers



Remark n = 0, 1

If the set value of the TABnCCRb register is smaller than the set value of the TABnCCR0 register, the INTTBnCCb signal is generated once per cycle.

Figure 7-19. Timing Chart When $D_{01} \geq D_{b1}$

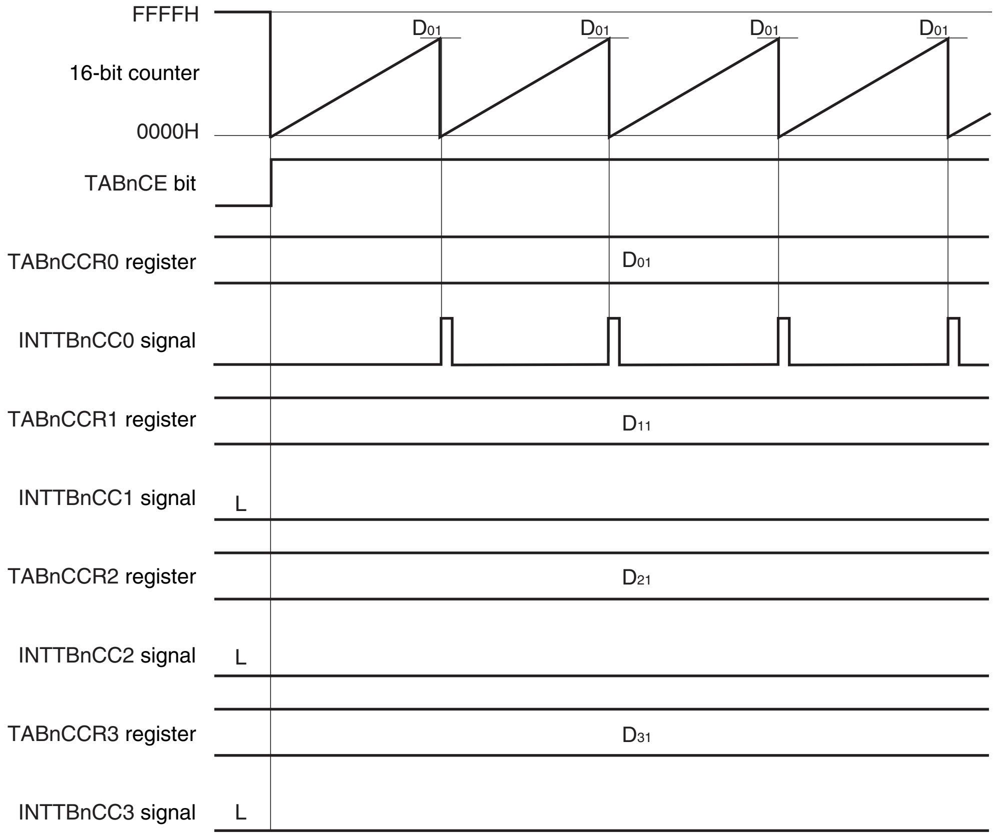


Remark n = 0, 1
b = 1 to 3

If the set value of the TABnCCRb register is greater than the set value of the TABnCCR0 register, the INTTBnCCb signal is not generated because the count value of the 16-bit counter and the value of the TABnCCRb register do not match.

When the TABnCCRb register is not used, it is recommended to set its value to FFFFH.

Figure 7-20. Timing Chart When $D_{01} < D_{b1}$



Remark n = 0, 1
b = 1 to 3

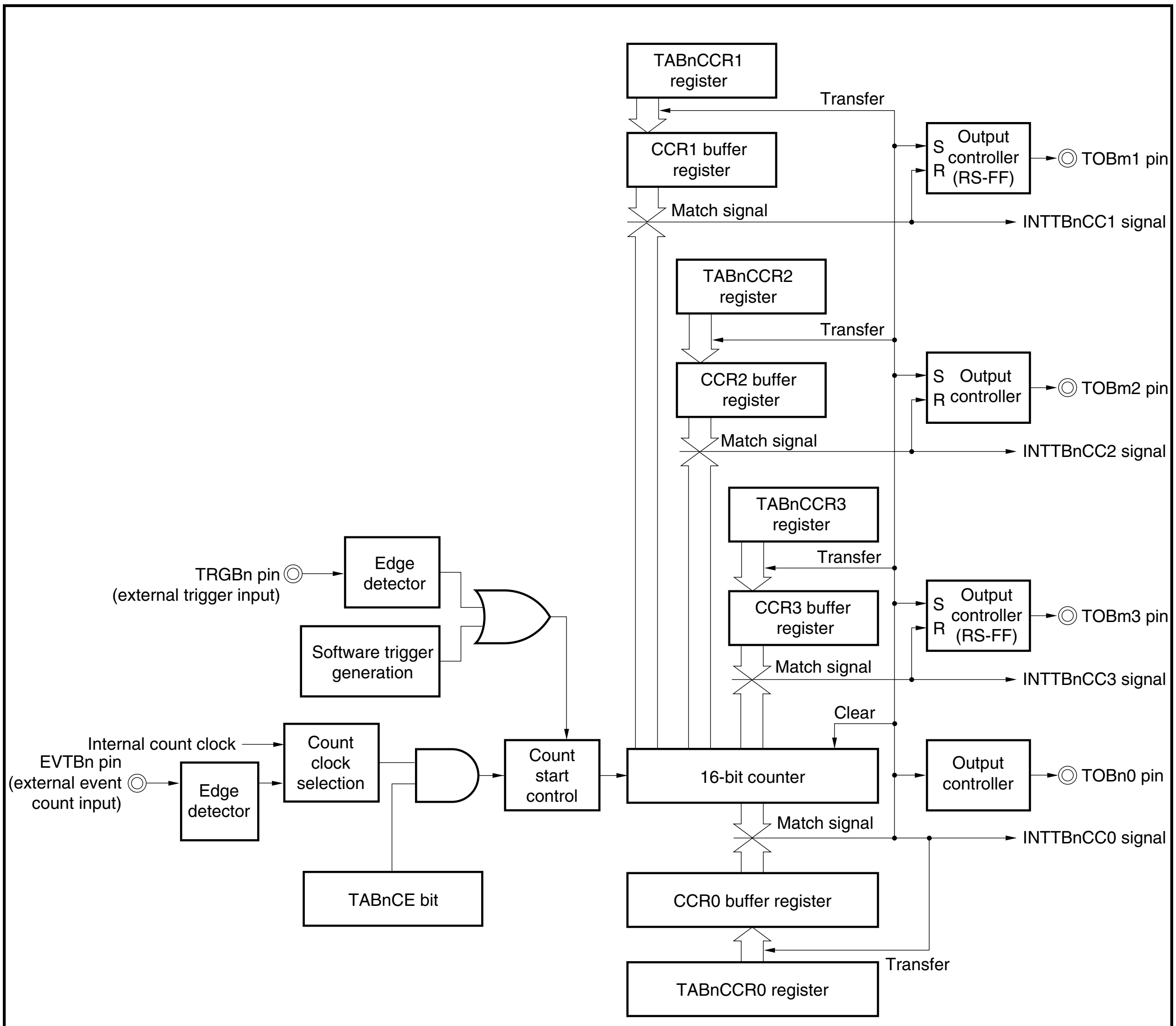
7.6.3 External trigger pulse output mode (TABnMD2 to TABnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter AB waits for a trigger when the TABnCTL0.TABnCE bit is set to 1. When the valid edge of an external trigger input signal (TRGBn) is detected, 16-bit timer/event counter AB starts counting, and outputs a PWM waveform (up to 3-phase) from the TOBm1 to TOBm3 pins. A PWM waveform with a duty factor of 50% whose half cycle is the set value of the TABnCCR0 register + 1 can also be output from the TOBn0 pin.

Pulses can also be output by generating a software trigger instead of using the external trigger input.

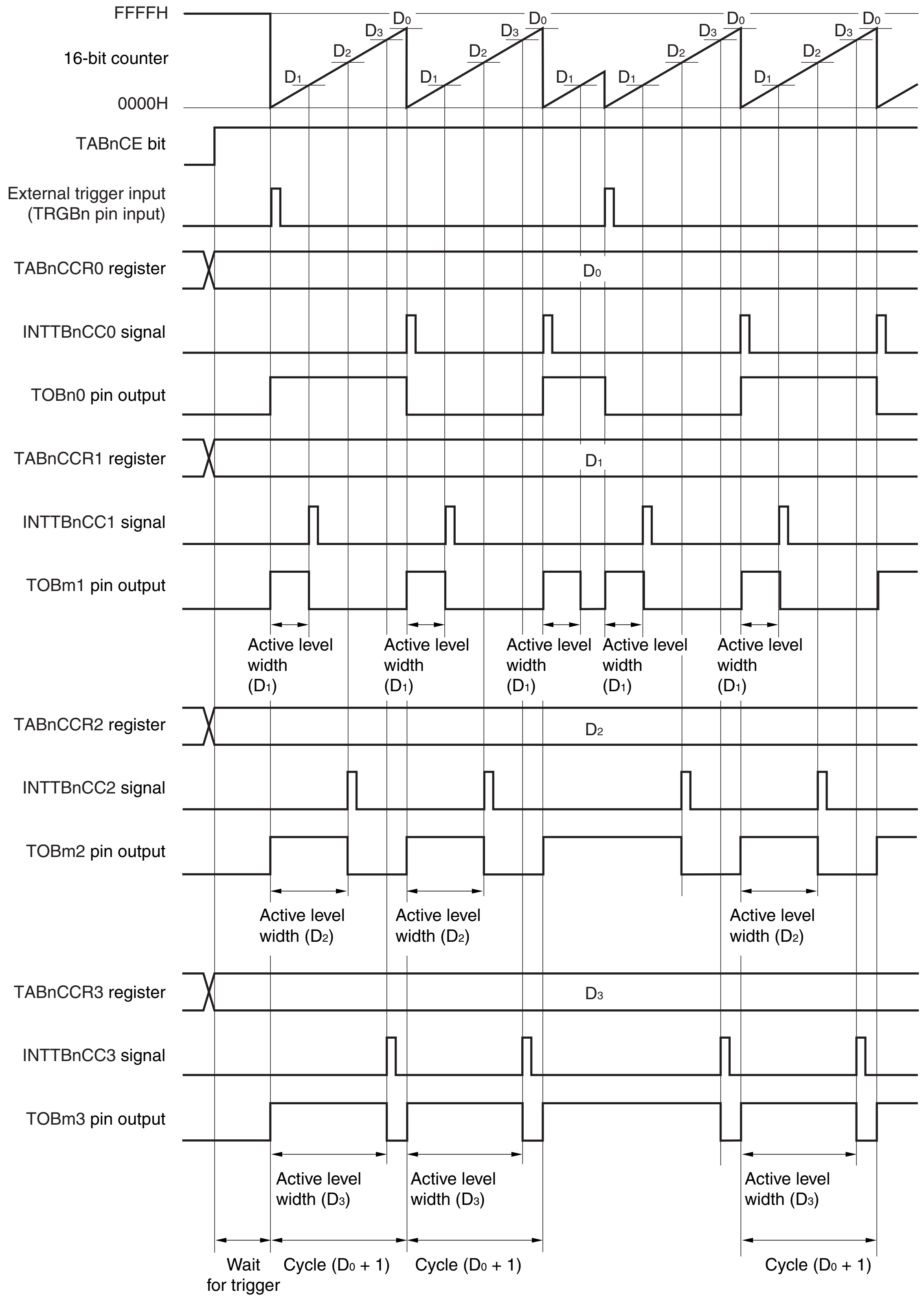
Caution The TAB1 output of the V850E/IG4-H is one PWM output with a duty factor of 50%

Figure 7-21. Configuration in External Trigger Pulse Output Mode



Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

Figure 7-22. Basic Timing in External Trigger Pulse Output Mode



Remark V850E/IG4-H: n = 0, 1, m = 0

V850E/IH4-H: n = 0, 1, m = 0, 1

16-bit timer/event counter AB waits for a trigger when the TABnCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TOBmb pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOBn0 pin is inverted. The TOBmb pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TABnCCRb register) × Count clock cycle
Cycle = (Set value of TABnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TABnCCRb register)/(Set value of TABnCCR0 register + 1)

The compare match request signal INTTBnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

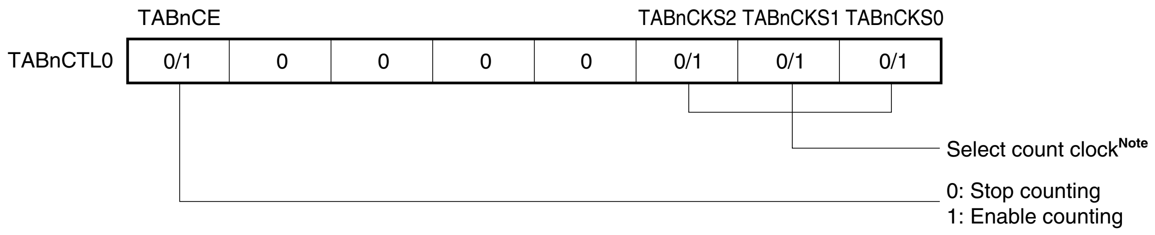
The value set to the TABnCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input signal (TRGBn), or setting the software trigger (TABnCTL1.TABnEST bit) to 1 is used as the trigger.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3, b = 1 to 3

Figure 7-23. Setting of Registers in External Trigger Pulse Output Mode (1/3)

(a) TABn control register 0 (TABnCTL0)



Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1.

Figure 7-23. Setting of Registers in External Trigger Pulse Output Mode (2/3)

(b) TABn control register 1 (TABnCTL1)

		TABnEST	TABnEEE			TABnMD2	TABnMD1	TABnMD0
TABnCTL1	0	0/1	0/1	0	0	0	1	0

TABnMD2 to TABnMD0 — 0, 1, 0: External trigger pulse output mode

TABnEEE — 0: Operate on count clock selected by TABnCKS0 to TABnCKS2 bits
1: Count with external event count input signal

TABnEST — Generate software trigger when 1 is written

(c) TABn I/O control register 0 (TABnIOC0)

	TABmOL3	TABmOE3	TABmOL2	TABmOE2	TABmOL1	TABmOE1	TABmOL0	TABmOE0
TABnIOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

TABmOE0 — 0: Disable TOBn0 pin output
1: Enable TOBn0 pin output

TABmOL0 — Setting of TOBn0 pin output level while waiting for external trigger
0: Low level
1: High level

TABmOE1 — 0: Disable TOBm1 pin output
1: Enable TOBm1 pin output

TABmOL1 — Setting of TOBm1 pin output level while waiting for external trigger
0: Low level
1: High level

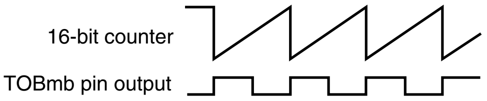
TABmOE2 — 0: Disable TOBm2 pin output
1: Enable TOBm2 pin output

TABmOL2 — Setting of TOBm2 pin output level while waiting for external trigger
0: Low level
1: High level

TABmOE3 — 0: Disable TOBm3 pin output
1: Enable TOBm3 pin output

TABmOL3 — Setting of TOBm3 pin output level while waiting for external trigger
0: Low level
1: High level

- When TABmOLb bit = 0



- When TABmOLb bit = 1

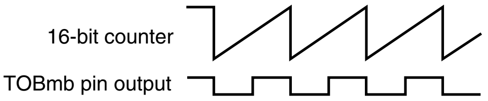
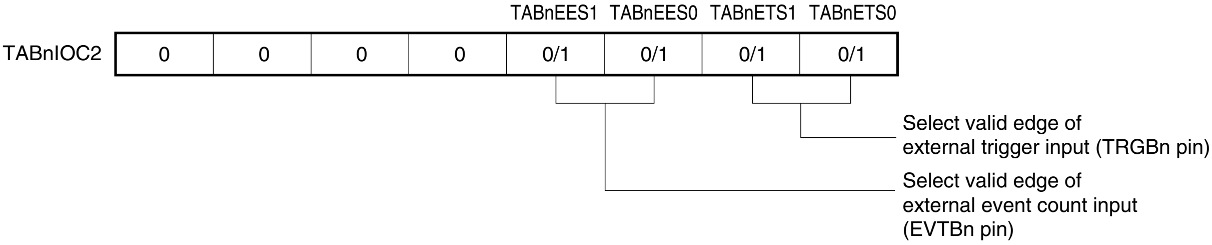


Figure 7-23. Setting of Registers in External Trigger Pulse Output Mode (3/3)

(d) TABn I/O control register 2 (TABnIOC2)



					TABnEES1	TABnEES0	TABnETS1	TABnETS0
TABnIOC2	0	0	0	0	0/1	0/1	0/1	0/1

Select valid edge of external trigger input (TRGBn pin)

Select valid edge of external event count input (EVTBn pin)

(e) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register, D_1 to the TABnCCR1 register, D_2 to the TABnCCR2 register, and D_3, to the TABnCCR3 register, the cycle and active level of the PWM waveform are as follows.

Cycle = (D_0 + 1) × Count clock cycle
TOBm1 pin PWM waveform active level width = D_1 × Count clock cycle
TOBm2 pin PWM waveform active level width = D_2 × Count clock cycle
TOBm3 pin PWM waveform active level width = D_3 × Count clock cycle

Remarks 1. TABm I/O control register 1 (TABmIOC1) and TABn option register 0 (TABnOPT0) are not used in the external trigger pulse output mode.

2. V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

(1) Operation flow in external trigger pulse output mode

Figure 7-24. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

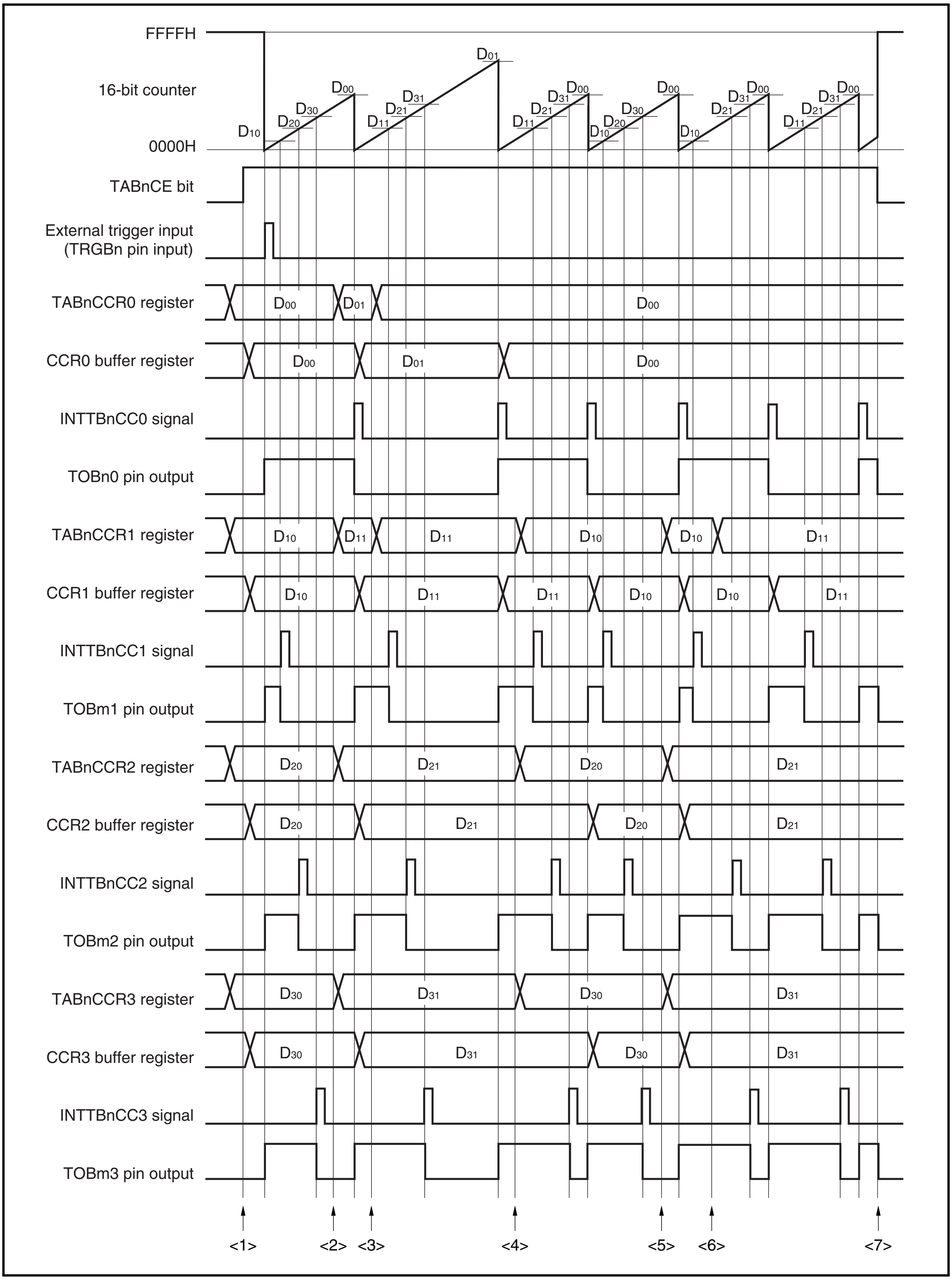
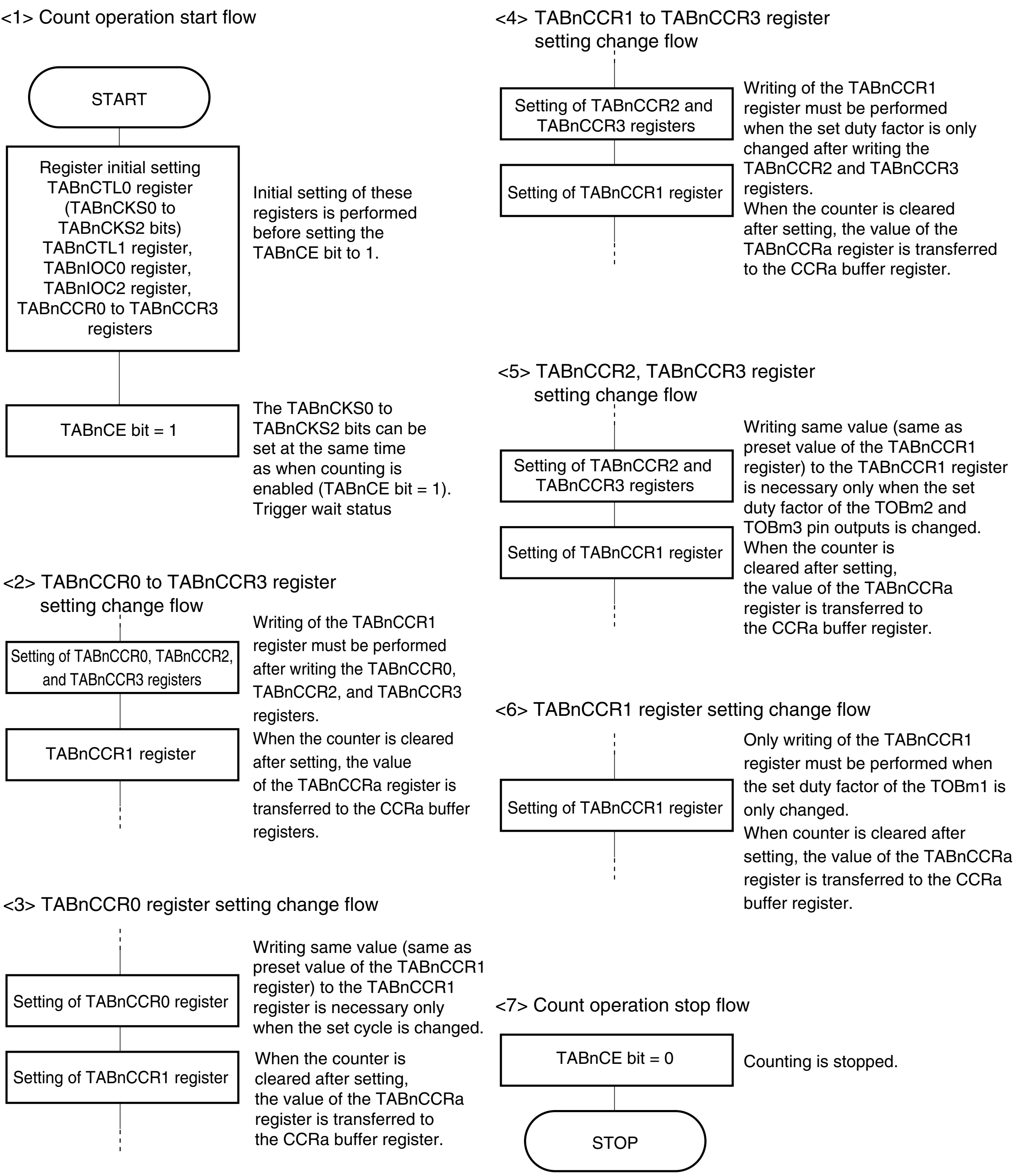


Figure 7-24. Software Processing Flow in External Trigger Pulse Output Mode (2/2)


START
Register initial setting TABnCTL0 register (TABnCKS0 to TABnCKS2 bits) TABnCTL1 register, TABnIOC0 register, TABnIOC2 register, TABnCCR0 to TABnCCR3 registers
TABnCE bit = 1
Setting of TABnCCR0, TABnCCR2, and TABnCCR3 registers
TABnCCR1 register
Setting of TABnCCR0 register
Setting of TABnCCR1 register
Setting of TABnCCR2 and TABnCCR3 registers
Setting of TABnCCR1 register
Setting of TABnCCR2 and TABnCCR3 registers
Setting of TABnCCR1 register
Setting of TABnCCR1 register
TABnCE bit = 0
STOP

<1> Count operation start flow

Initial setting of these registers is performed before setting the TABnCE bit to 1.

The TABnCKS0 to TABnCKS2 bits can be set at the same time as when counting is enabled (TABnCE bit = 1). Trigger wait status

<2> TABnCCR0 to TABnCCR3 register setting change flow

Writing of the TABnCCR1 register must be performed after writing the TABnCCR0, TABnCCR2, and TABnCCR3 registers. When the counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer registers.

<3> TABnCCR0 register setting change flow

Writing same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register is necessary only when the set cycle is changed.

When the counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer register.

<4> TABnCCR1 to TABnCCR3 register setting change flow

Writing of the TABnCCR1 register must be performed when the set duty factor is only changed after writing the TABnCCR2 and TABnCCR3 registers. When the counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer register.

<5> TABnCCR2, TABnCCR3 register setting change flow

Writing same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register is necessary only when the set duty factor of the TOBm2 and TOBm3 pin outputs is changed. When the counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer register.

<6> TABnCCR1 register setting change flow

Only writing of the TABnCCR1 register must be performed when the set duty factor of the TOBm1 is only changed. When counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer register.

<7> Count operation stop flow

Counting is stopped.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3

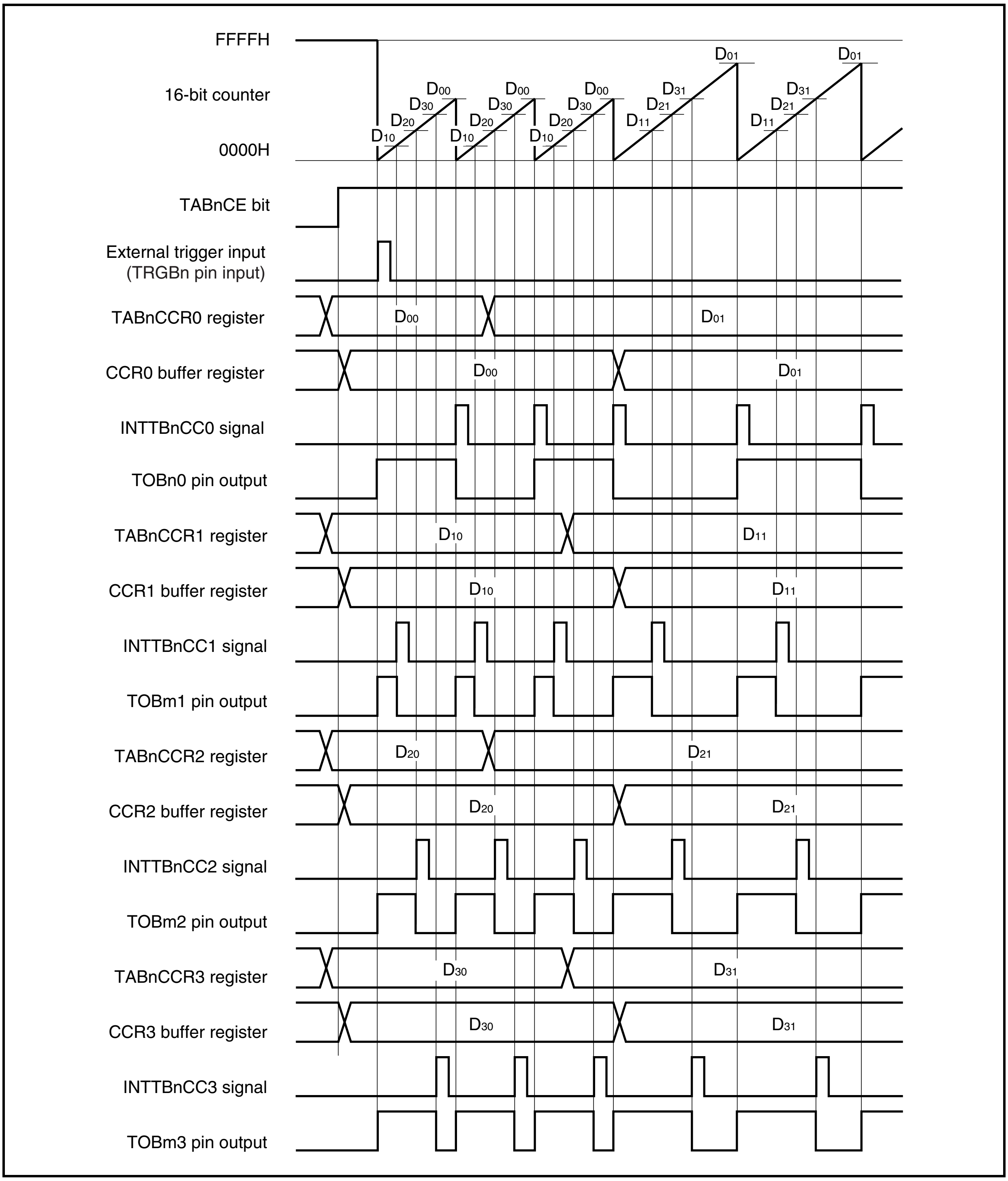
(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TABnCCR1 register last.

Rewrite the TABnCCRb register after writing the TABnCCR1 register after the INTTBnCC0 signal is detected.

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3



In order to transfer data from the TABnCCRa register to the CCRa buffer register, the TABnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TABnCCR0 register, set the active level width to the TABnCCR2 and TABnCCR3 registers, and then set an active level to the TABnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TABnCCR0 register, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, first set an active level to the TABnCCR2 and TABnCCR3 registers and then set an active level to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOBm1 pin, only the TABnCCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOBm2 and TOBm3 pins, first set an active level width to the TABnCCR2 and TABnCCR3 registers, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

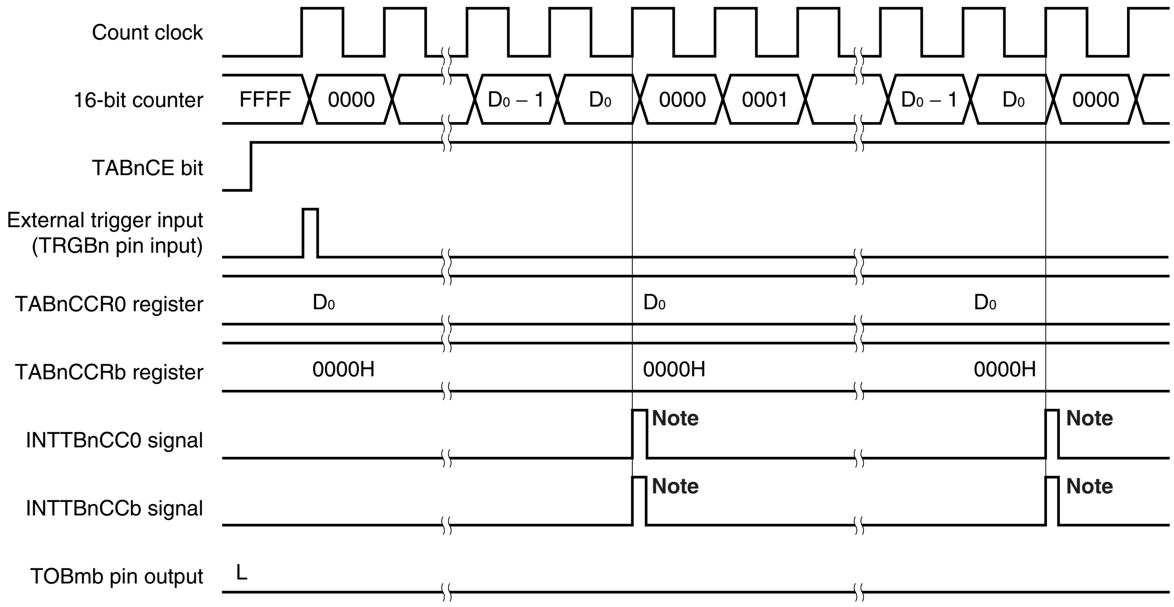
After data is written to the TABnCCR1 register, the value written to the TABnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TABnCCR0 to TABnCCR3 registers again after writing the TABnCCR1 register once, do so after the INTTBnCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because timing of transferring data from the TABnCCRa register to the CCRa buffer register conflicts with writing the TABnCCRa register.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3

(b) 0%/100% output of PWM waveform

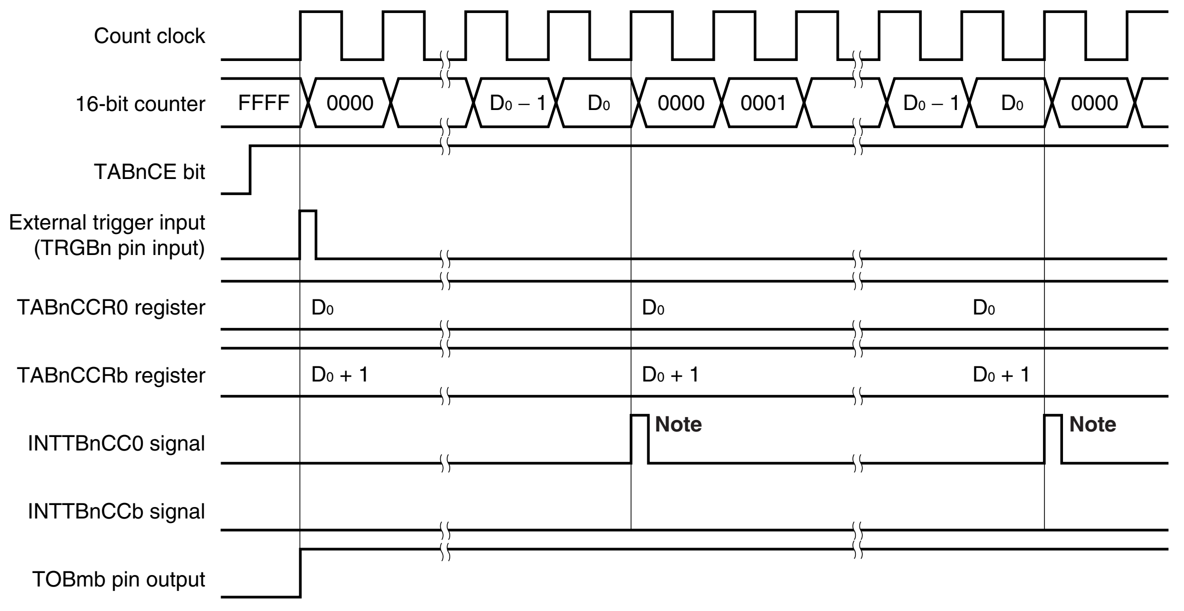
To output a 0% waveform, set the TABnCCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTTBnCC0 and INTTBnCCb signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (f_{XX}).

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

To output a 100% waveform, set a value of (set value of TABnCCR0 register + 1) to the TABnCCRb register. If the set value of the TABnCCR0 register is FFFFH, 100% output cannot be produced.

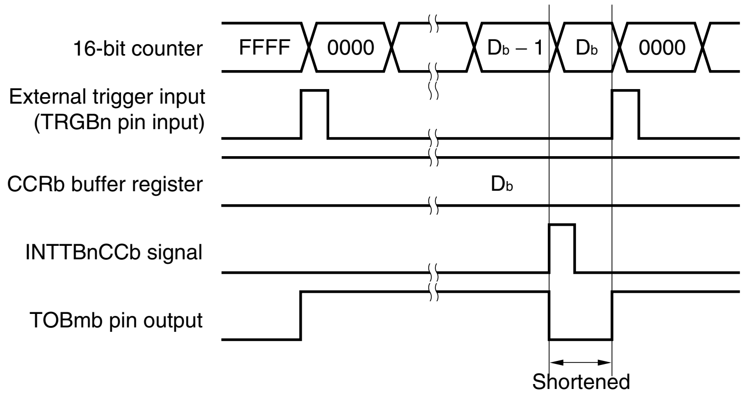


Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

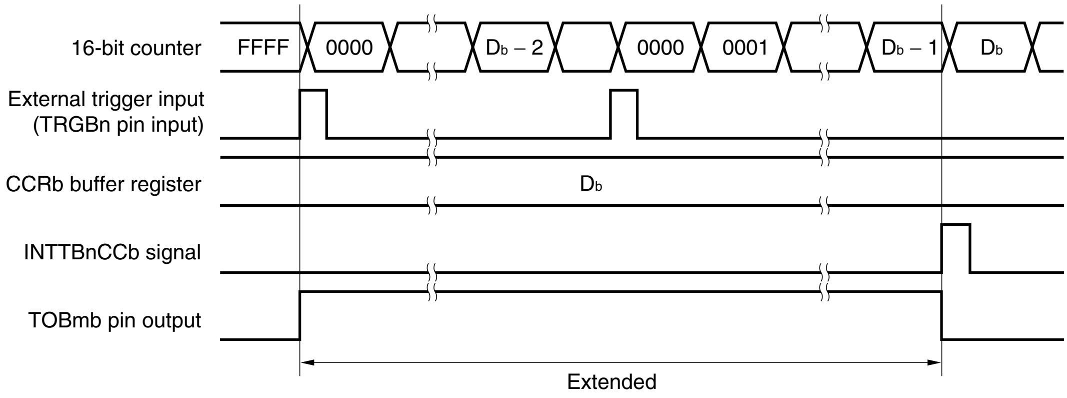
(c) Conflict between trigger detection and match with CCRb buffer register

If the trigger is detected immediately after the INTTBnCCb signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOBmb pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

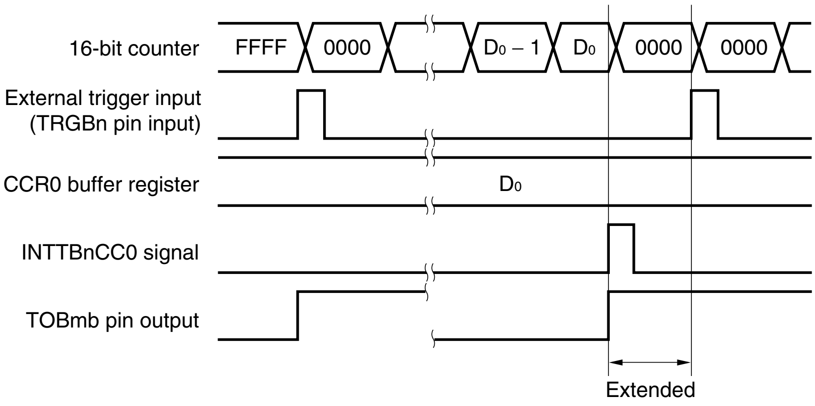
If the trigger is detected immediately before the INTTBnCCb signal is generated, the INTTBnCCb signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOBmb pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

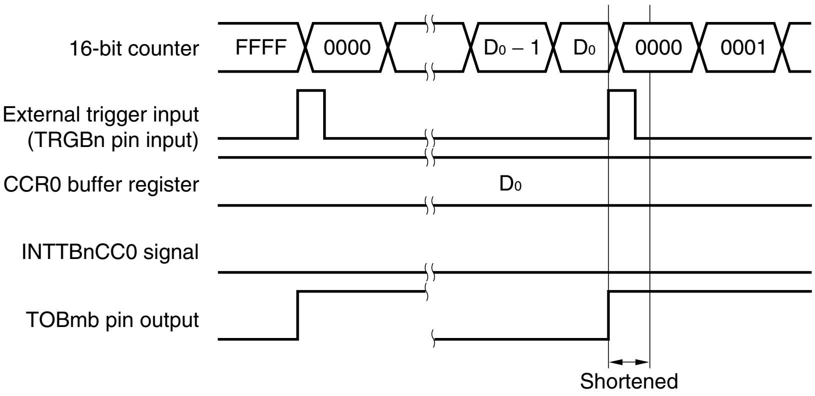
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTTBnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOBmb pin is extended by time from generation of the INTTBnCC0 signal to trigger detection.



Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

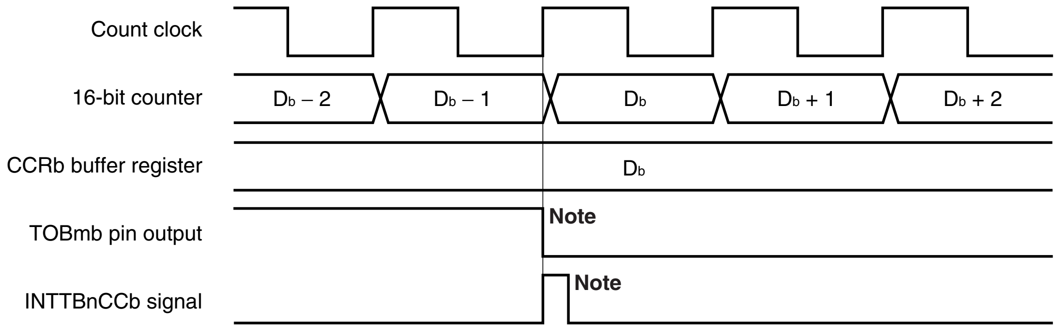
If the trigger is detected immediately before the INTTBnCC0 signal is generated, the INTTBnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOBmb pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

(e) Generation timing of compare match interrupt request signal (INTTBnCCb)

The timing of generation of the INTTBnCCb signal in the external trigger pulse output mode differs from the timing of INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.



Note The timing is actually delayed by one operating clock (f_{XX}).

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

Usually, the INTTBnCCb signal is generated in synchronization with the next count up after the count value of the 16-bit counter matches the value of the CCRb buffer register.

In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOBmb pin.

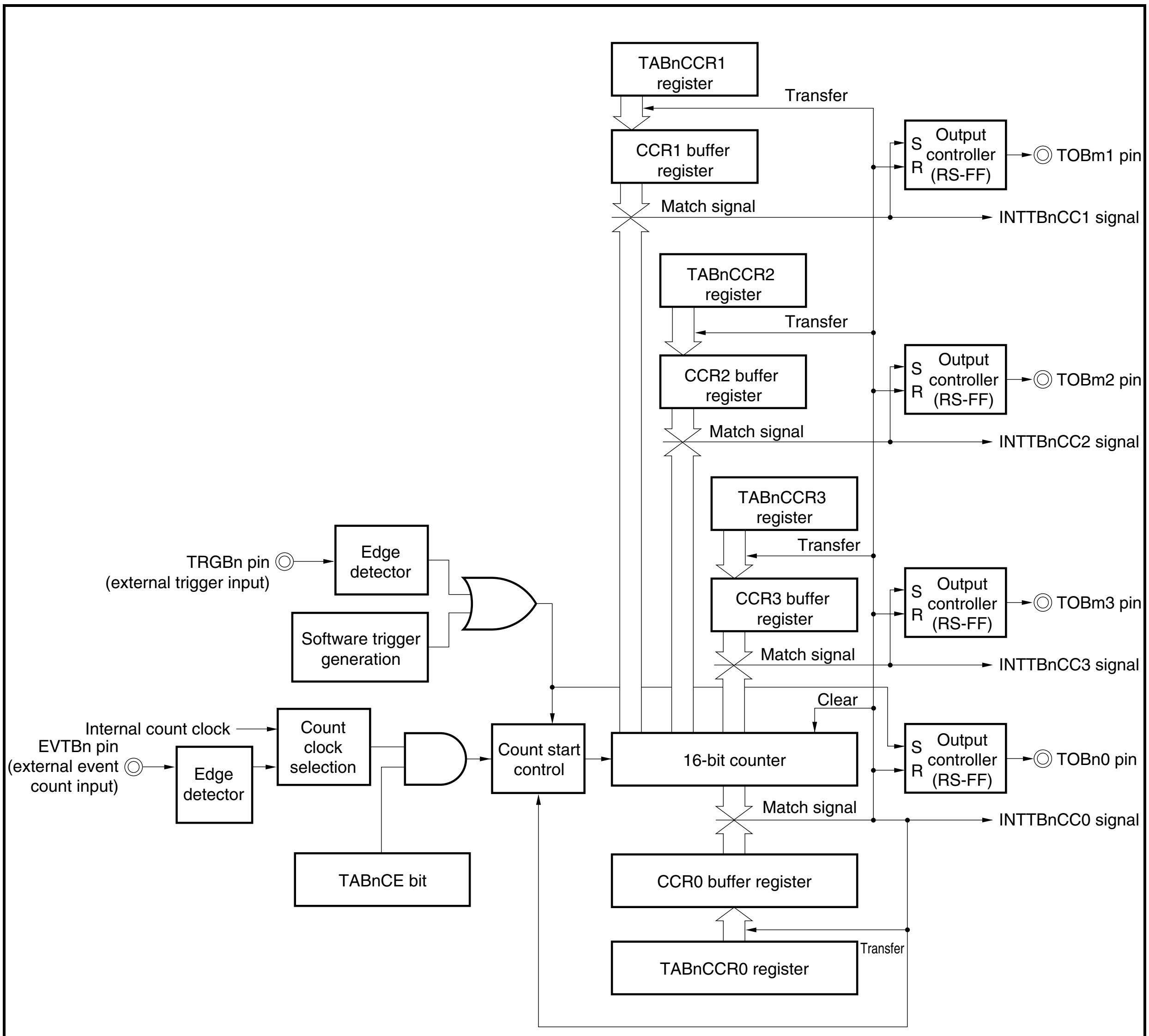
7.6.4 One-shot pulse output mode (TABnMD2 to TABnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter AB waits for a trigger when the TABnCTL0.TABnCE bit is set to 1. When the valid edge of an external trigger input (TRGBn) is detected, 16-bit timer/event counter AB starts counting, and outputs a one-shot pulse from the TOBm1 to TOBm3 pins. The TOBn0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

Instead of the external trigger input, a software trigger can also be generated to output the pulse.

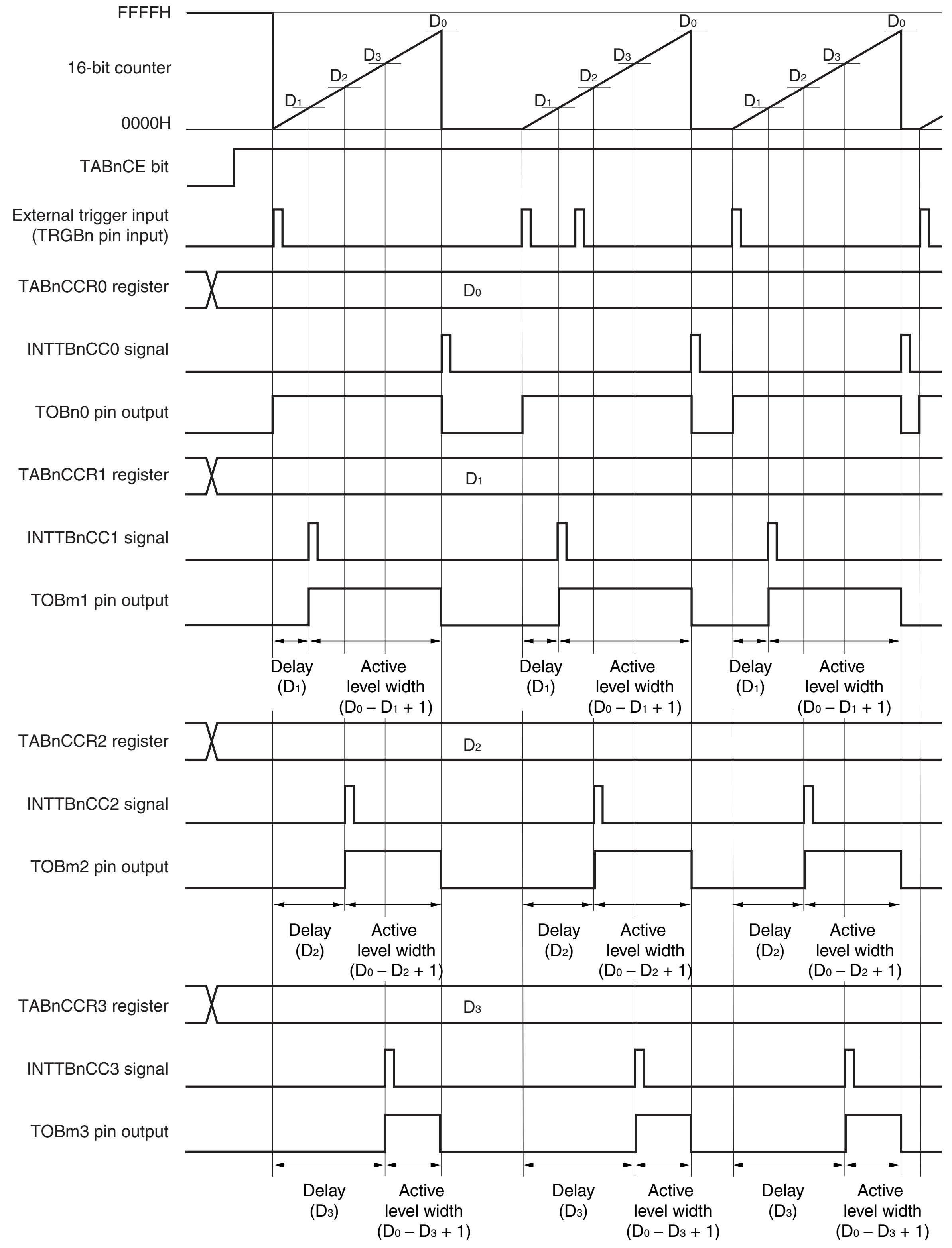
Caution The TAB1 output of the V850E/IG4-H is one PWM output.

Figure 7-25. Configuration in One-Shot Pulse Output Mode



Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

Figure 7-26. Basic Timing in One-Shot Pulse Output Mode



Remark V850E/IG4-H: n = 0, 1, m = 0

V850E/IH4-H: n = 0, 1, m = 0, 1

When the TABnCE bit is set to 1, 16-bit timer/event counter AB waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOBmb pin. After the one-shot pulse is output, the 16-bit counter is set to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

Output delay period = (Set value of TABnCCRb register) × Count clock cycle

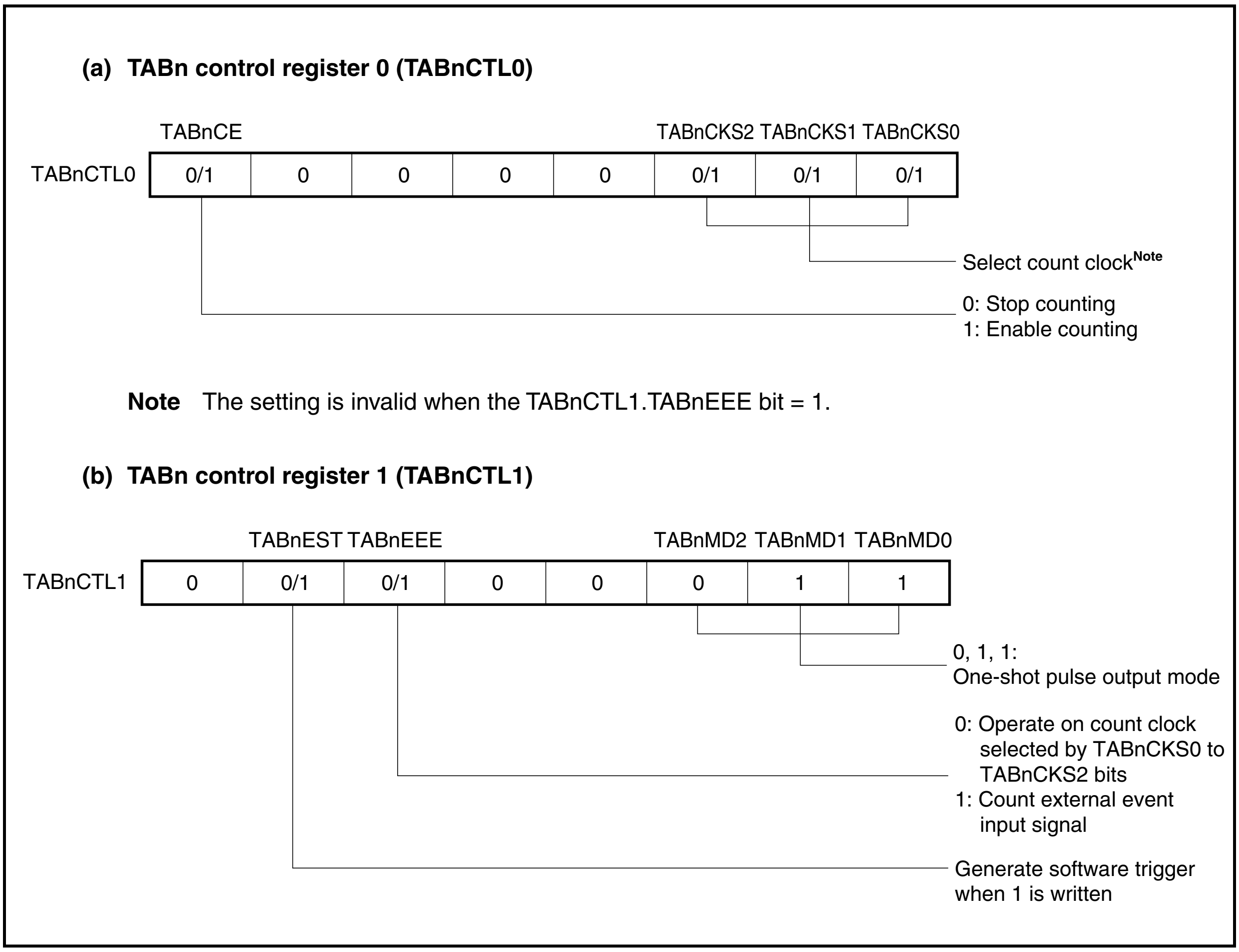
Active level width = (Set value of TABnCCR0 register − Set value of TABnCCRb register + 1) × Count clock cycle

The compare match interrupt request signal INTTBnCC0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

The valid edge of an external trigger input (TRGBn) or setting the software trigger (TABnCTL1.TABnEST bit) to 1 is used as the trigger.

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

Figure 7-27. Setting of Registers in One-Shot Pulse Output Mode (1/3)



(a) TABn control register 0 (TABnCTL0)

	TABnCE					TABnCKS2	TABnCKS1	TABnCKS0
TABnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clockNote
0: Stop counting
1: Enable counting

Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1.

(b) TABn control register 1 (TABnCTL1)

		TABnEST	TABnEEE			TABnMD2	TABnMD1	TABnMD0
TABnCTL1	0	0/1	0/1	0	0	0	1	1

0, 1, 1:
One-shot pulse output mode
0: Operate on count clock selected by TABnCKS0 to TABnCKS2 bits
1: Count external event input signal
Generate software trigger when 1 is written

Figure 7-27. Register Setting in One-Shot Pulse Output Mode (2/3)

(c) TABn I/O control register 0 (TABnIOC0)

	TABmOL3	TABmOE3	TABmOL2	TABmOE2	TABmOL1	TABmOE1	TABmOL0	TABmOE0
TABnIOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

TABmOE0:
0: Disable TOBn0 pin output
1: Enable TOBn0 pin output

TABmOL0:
Setting of TOBn0 pin output level while waiting for external trigger
0: Low level
1: High level

TABmOE1:
0: Disable TOBm1 pin output
1: Enable TOBm1 pin output

TABmOL1:
Setting of TOBm1 pin output level while waiting for external trigger
0: Low level
1: High level

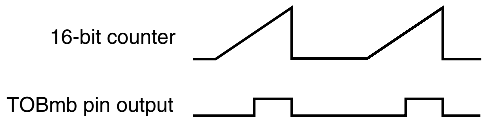
TABmOE2:
0: Disable TOBm2 pin output
1: Enable TOBm2 pin output

TABmOL2:
Setting of TOBm2 pin output level while waiting for external trigger
0: Low level
1: High level

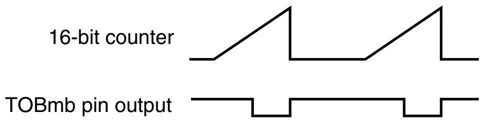
TABmOE3:
0: Disable TOBm3 pin output
1: Enable TOBm3 pin output

TABmOL3:
Setting of TOBm3 pin output level while waiting for external trigger
0: Low level
1: High level

- When TABmOLb bit = 0



- When TABmOLb bit = 1



(d) TABn I/O control register 2 (TABnIOC2)

					TABnEES1	TABnEES0	TABnETS1	TABnETS0
TABnIOC2	0	0	0	0	0/1	0/1	0/1	0/1

TABnETS1, TABnETS0: Select valid edge of external trigger input (TRGBn pin)

TABnEES1, TABnEES0: Select valid edge of external event count input (EVTBn pin)

(e) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

Figure 7-27. Register Setting in One-Shot Pulse Output Mode (3/3)

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register and D_b to the TABnCCRb register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width = (D_b – D_0 + 1) × Count clock cycle

Output delay period = D_b × Count clock cycle

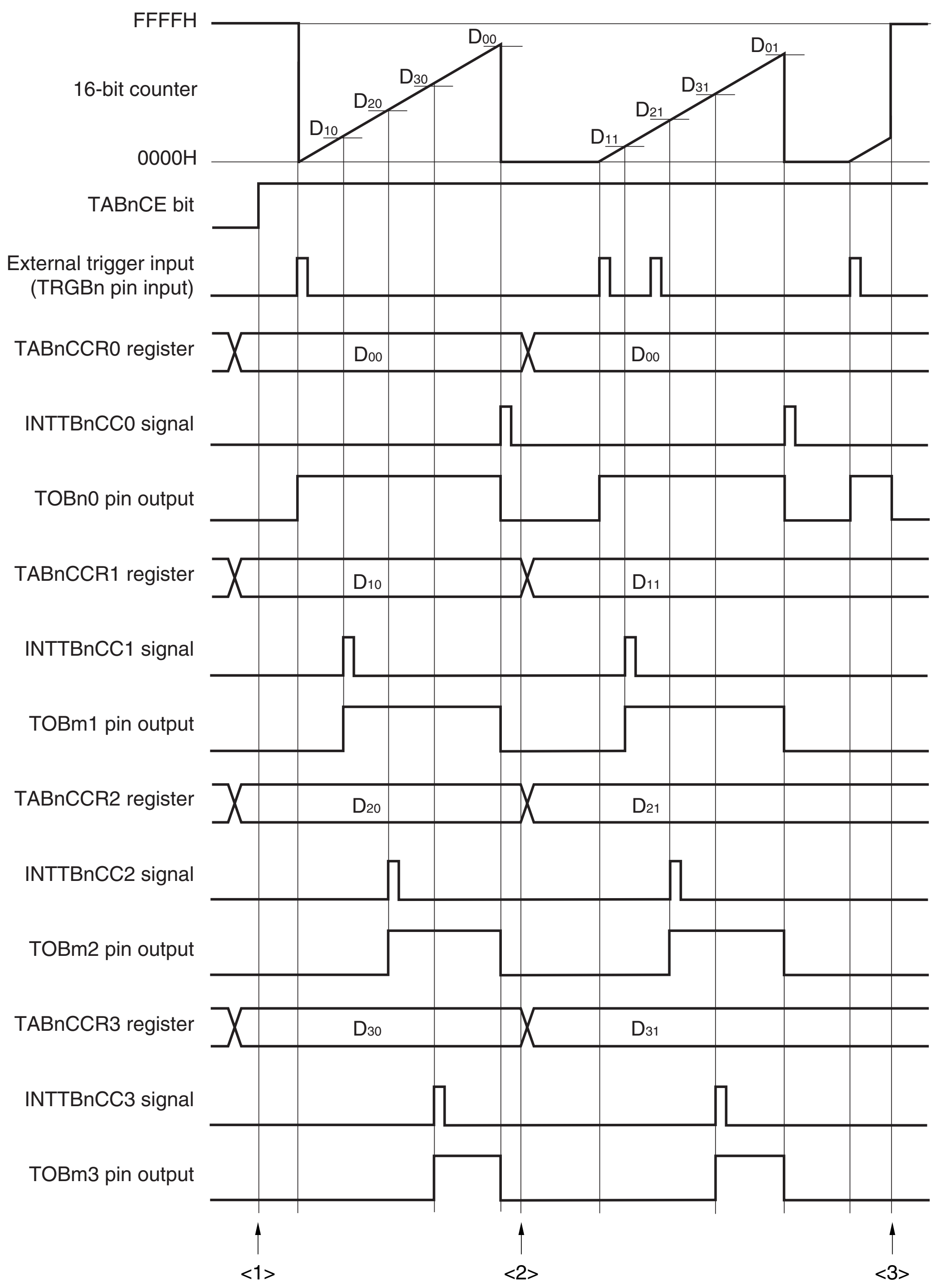
Caution One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TABnCCRb register is greater than that set in the TABnCCR0 register.

Remarks 1. TABm I/O control register 1 (TABmIOC1) and TABn option register 0 (TABnOPT0) are not used in the one-shot pulse output mode.

2. V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

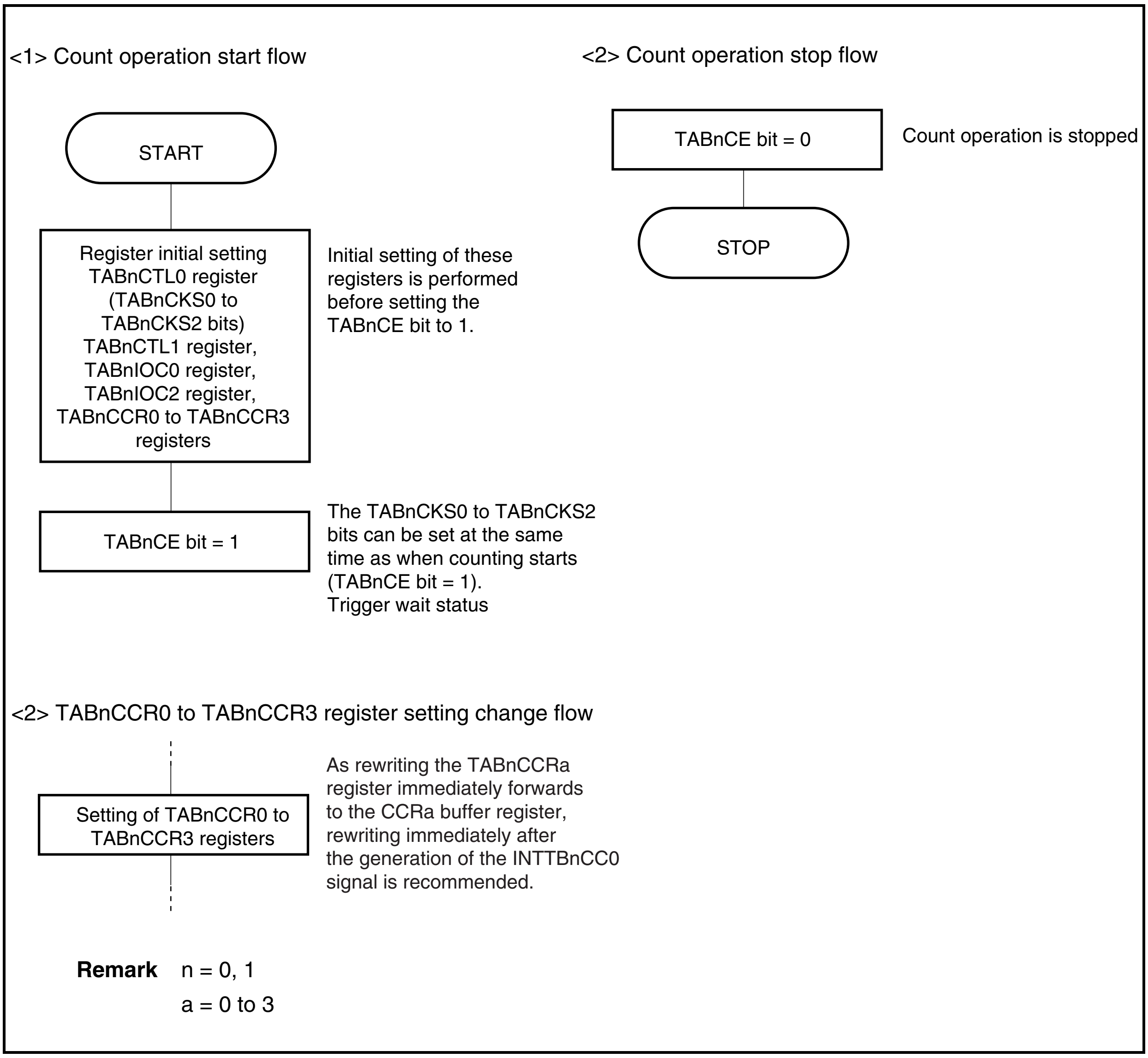
(1) Operation flow in one-shot pulse output mode

Figure 7-28. Software Processing Flow in One-Shot Pulse Output Mode (1/2)



Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

Figure 7-28. Software Processing Flow in One-Shot Pulse Output Mode (2/2)



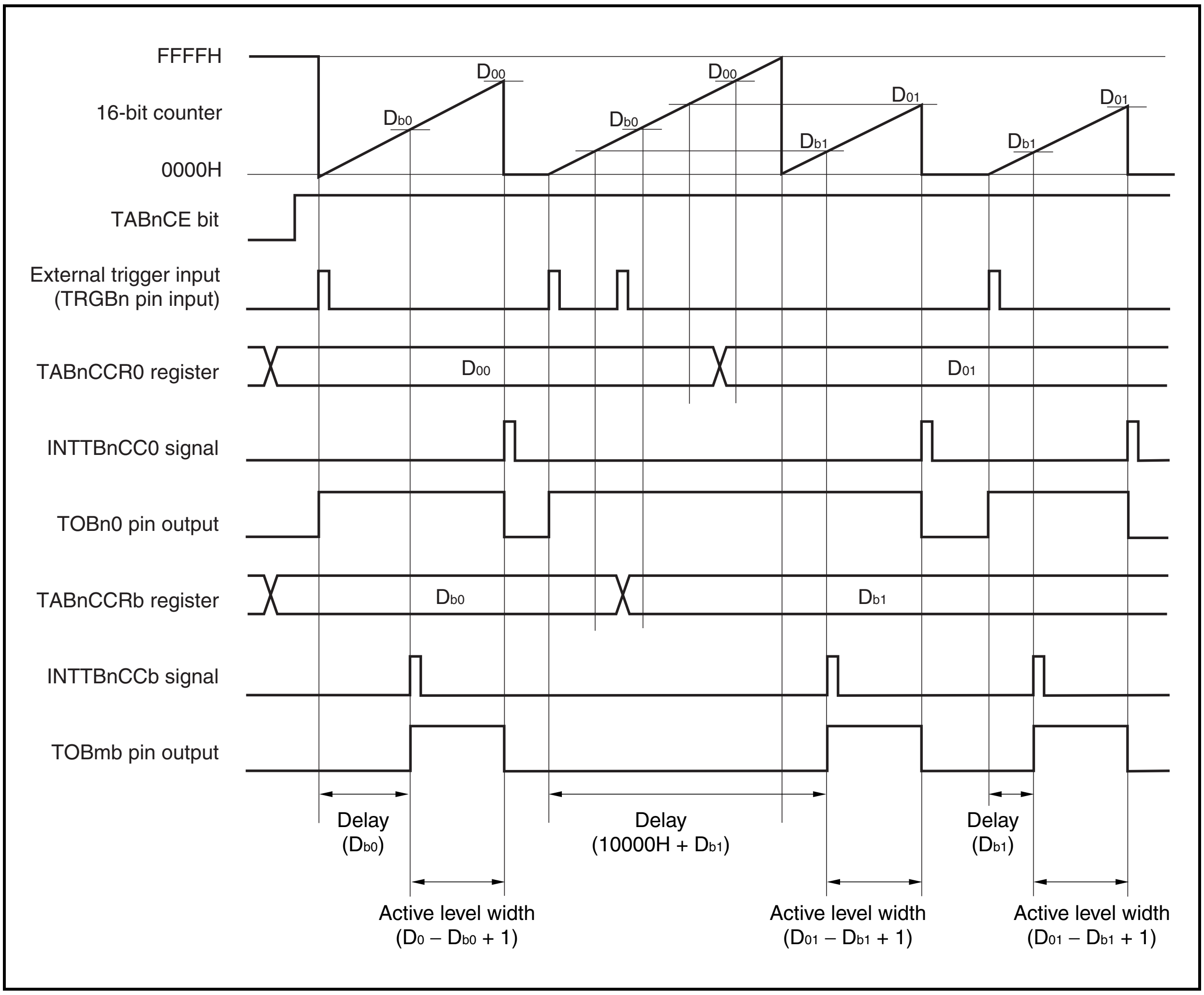
<1> Count operation start flow
START
Register initial setting TABnCTL0 register (TABnCKS0 to TABnCKS2 bits) TABnCTL1 register, TABnIOC0 register, TABnIOC2 register, TABnCCR0 to TABnCCR3 registers
Initial setting of these registers is performed before setting the TABnCE bit to 1.
TABnCE bit = 1
The TABnCKS0 to TABnCKS2 bits can be set at the same time as when counting starts (TABnCE bit = 1).
Trigger wait status
<2> Count operation stop flow
TABnCE bit = 0
Count operation is stopped
STOP
<2> TABnCCR0 to TABnCCR3 register setting change flow
Setting of TABnCCR0 to TABnCCR3 registers
As rewriting the TABnCCRa register immediately forwards to the CCRa buffer register, rewriting immediately after the generation of the INTTBnCC0 signal is recommended.

Remark n = 0, 1
a = 0 to 3

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TABnCCRa register

To change the set value of the TABnCCRa register to a smaller value, stop counting once, and then change the set value. When the overflow may occur, stop counting once, and then change the set value.



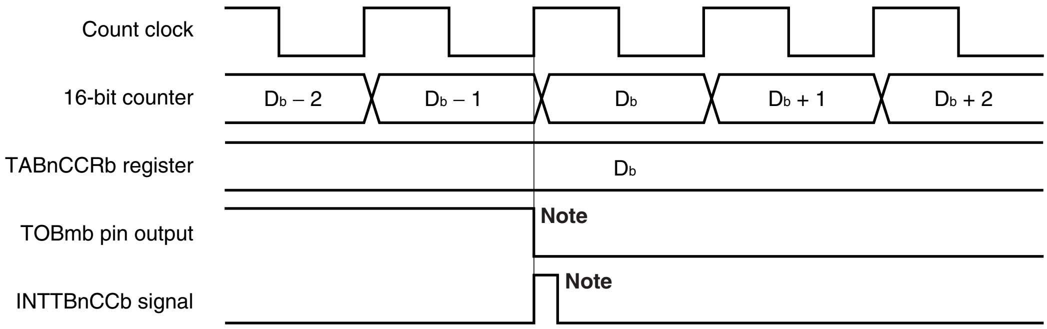
When the TABnCCR0 register is rewritten from D_{00} to D_{01} and the TABnCCRb register from D_{b0} to D_{b1} where $D_{00} > D_{01}$ and $D_{b0} > D_{b1}$, if the TABnCCRb register is rewritten when the count value of the 16-bit counter is greater than D_{b1} and less than D_{b0} and if the TABnCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{b1}, the counter generates the INTTBnCCb signal and asserts the TOBmb pin. When the count value matches D_{01}, the counter generates the INTTBnCC0 signal, deasserts the TOBmb pin, and stops counting.

Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3, b = 1 to 3

(b) Generation timing of compare match interrupt request signal (INTTBnCCb)

The generation timing of the INTTBnCCb signal in the one-shot pulse output mode is different from INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the TABnCCRb register.



Note The timing is actually delayed by one operating clock (f_{xx}).

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

Usually, the INTTBnCCb signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TABnCCRb register.

In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOBmb pin.

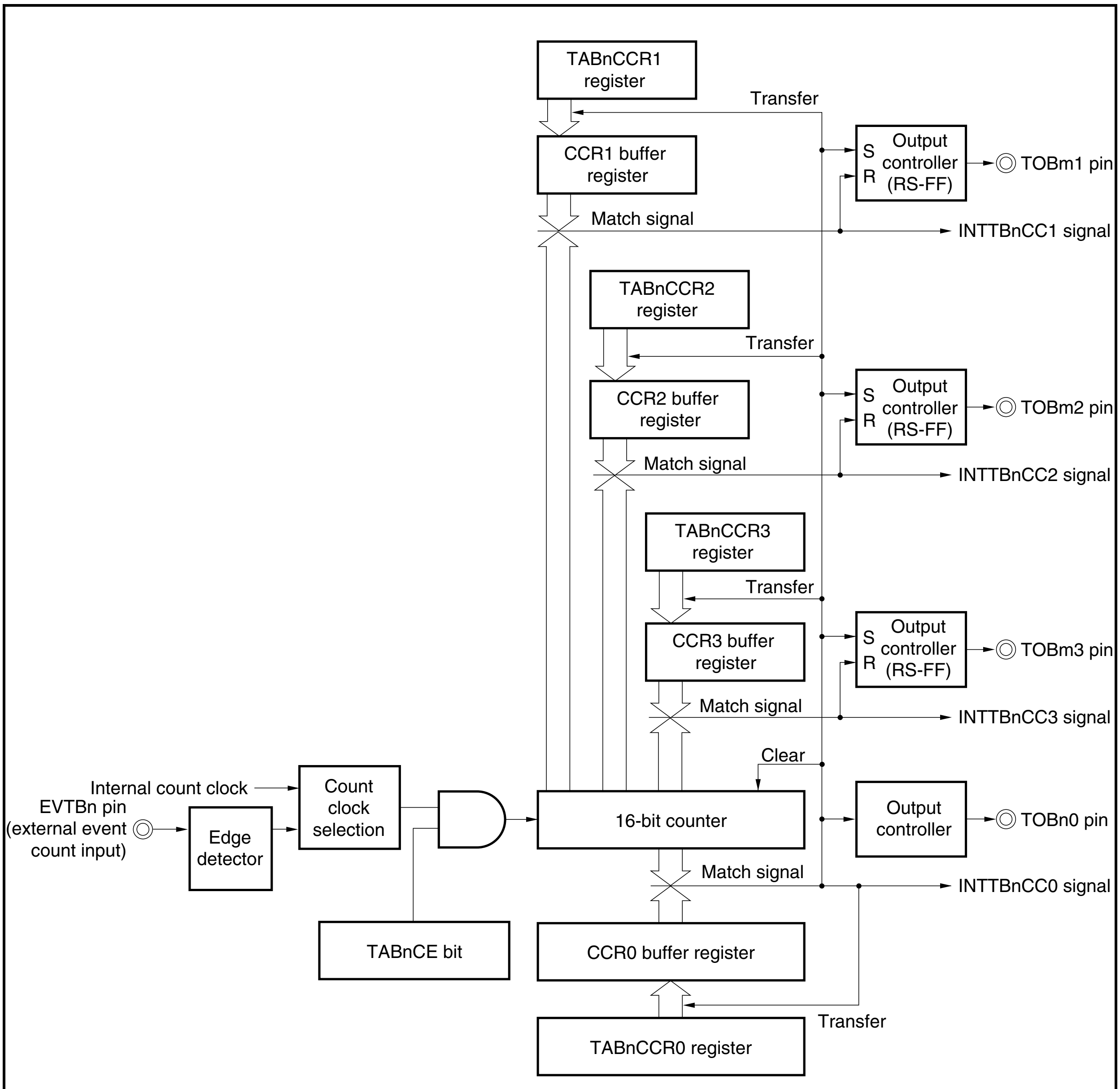
7.6.5 PWM output mode (TABnMD2 to TABnMD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TOBm1 to TOBm3 pins when the TABnCTL0.TABnCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TABnCCR0 register + 1 as half its cycle is output from the TOBn0 pin.

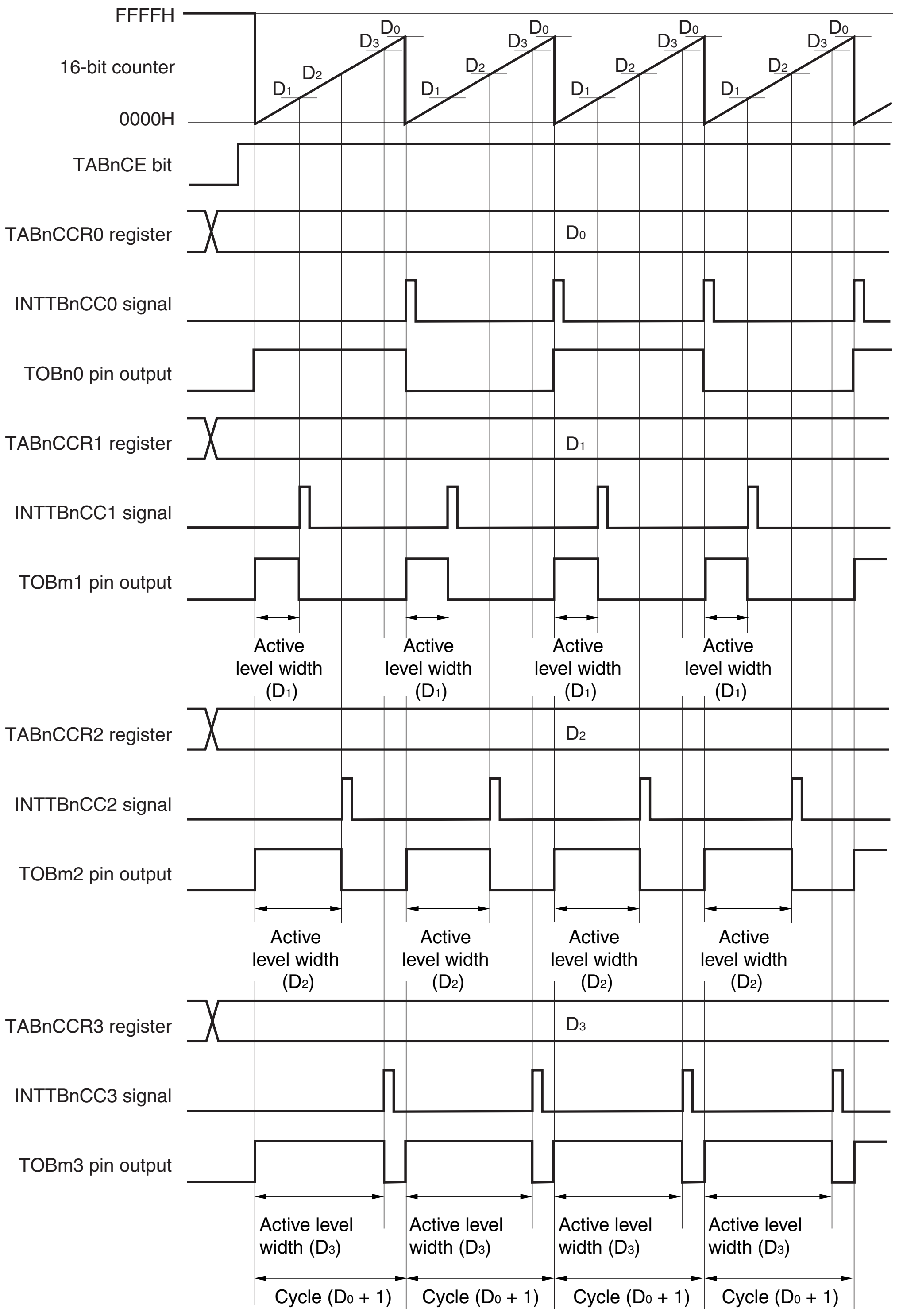
Caution The TAB1 output of the V850E/IG4-H is one PWM output with a duty factor of 50%

Figure 7-29. Configuration in PWM Output Mode



Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

Figure 7-30. Basic Timing in PWM Output Mode



Remark V850E/IG4-H: n = 0, 1, m = 0

V850E/IH4-H: n = 0, 1, m = 0, 1

When the TABnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs PWM waveform from the TOBmb pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

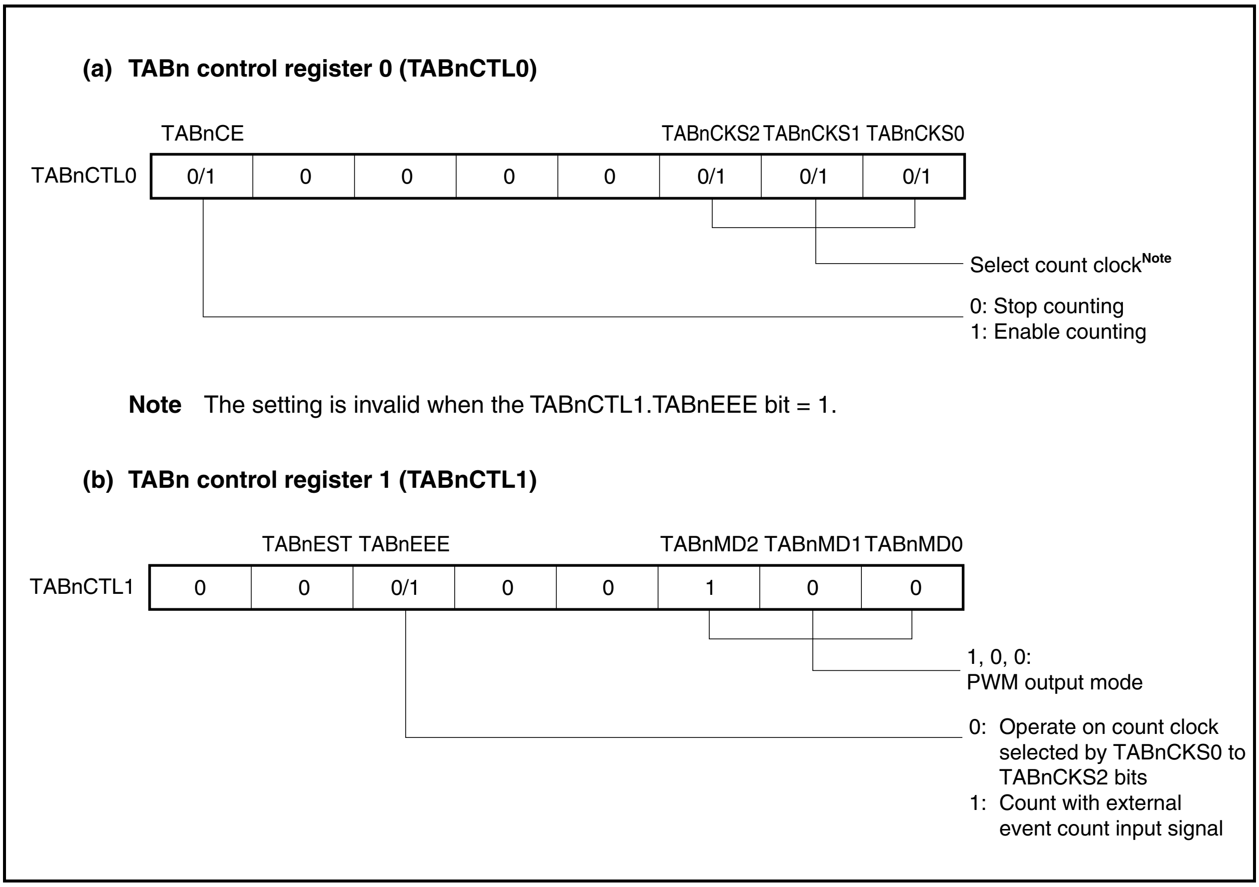
Active level width = (Set value of TABnCCRb register) × Count clock cycle
Cycle = (Set value of TABnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TABnCCRb register)/(Set value of TABnCCR0 register + 1)

The PWM waveform can be changed by rewriting the TABnCCRa register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal INTTBnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3, b = 1 to 3

Figure 7-31. Setting of Registers in PWM Output Mode (1/3)



(a) TABn control register 0 (TABnCTL0)

Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1.

(b) TABn control register 1 (TABnCTL1)

TABnEST
TABnEEE
TABnMD2
TABnMD1
TABnMD0
TABnCTL1
0
0
0/1
0
0
1
0
0
1, 0, 0:
PWM output mode
0: Operate on count clock selected by TABnCKS0 to TABnCKS2 bits
1: Count with external event count input signal

Figure 7-31. Setting of Registers in PWM Output Mode (2/3)

(c) TABn I/O control register 0 (TABnIOC0)

	TABmOL3	TABmOE3	TABmOL2	TABmOE2	TABmOL1	TABmOE1	TABmOL0	TABmOE0
TABnIOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

TABmOE0:
0: Disable TOBn0 pin output
1: Enable TOBn0 pin output

TABmOL0:
Setting of TOBn0 pin output level before count operation
0: Low level
1: High level

TABmOE1:
0: Disable TOBm1 pin output
1: Enable TOBm1 pin output

TABmOL1:
Setting of TOBm1 pin output level before count operation
0: Low level
1: High level

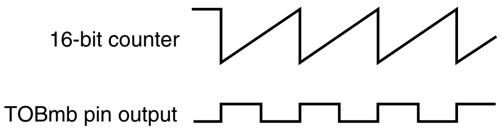
TABmOE2:
0: Disable TOBm2 pin output
1: Enable TOBm2 pin output

TABmOL2:
Setting of TOBm2 pin output level before count operation
0: Low level
1: High level

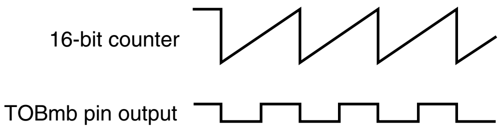
TABmOE3:
0: Disable TOBm3 pin output
1: Enable TOBm3 pin output

TABmOL3:
Setting of TOBm3 pin output level before count operation
0: Low level
1: High level

- When TABmOLb bit = 0



- When TABmOLb bit = 1



(d) TABn I/O control register 2 (TABnIOC2)

					TABnEES1	TABnEES0	TABnETS1	TABnETS0
TABnIOC2	0	0	0	0	0/1	0/1	0	0

TABnEES1, TABnEES0:
Select valid edge of external event count input (EVTBn pin).

(e) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

Figure 7-31. Register Setting in PWM Output Mode (3/3)

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register and D_b to the TABnCCRb register, the cycle and active level of the PWM waveform are as follows.

PWM waveform cycle = (D_0 + 1) × Count clock cycle
PWM waveform active level width = D_b × Count clock cycle

Remarks 1. TABm I/O control register 1 (TABmIOC1) and TABn option register 0 (TABnOPT0) are not used in the PWM output mode.

2. V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

(1) Operation flow in PWM output mode

Figure 7-32. Software Processing Flow in PWM Output Mode (1/2)

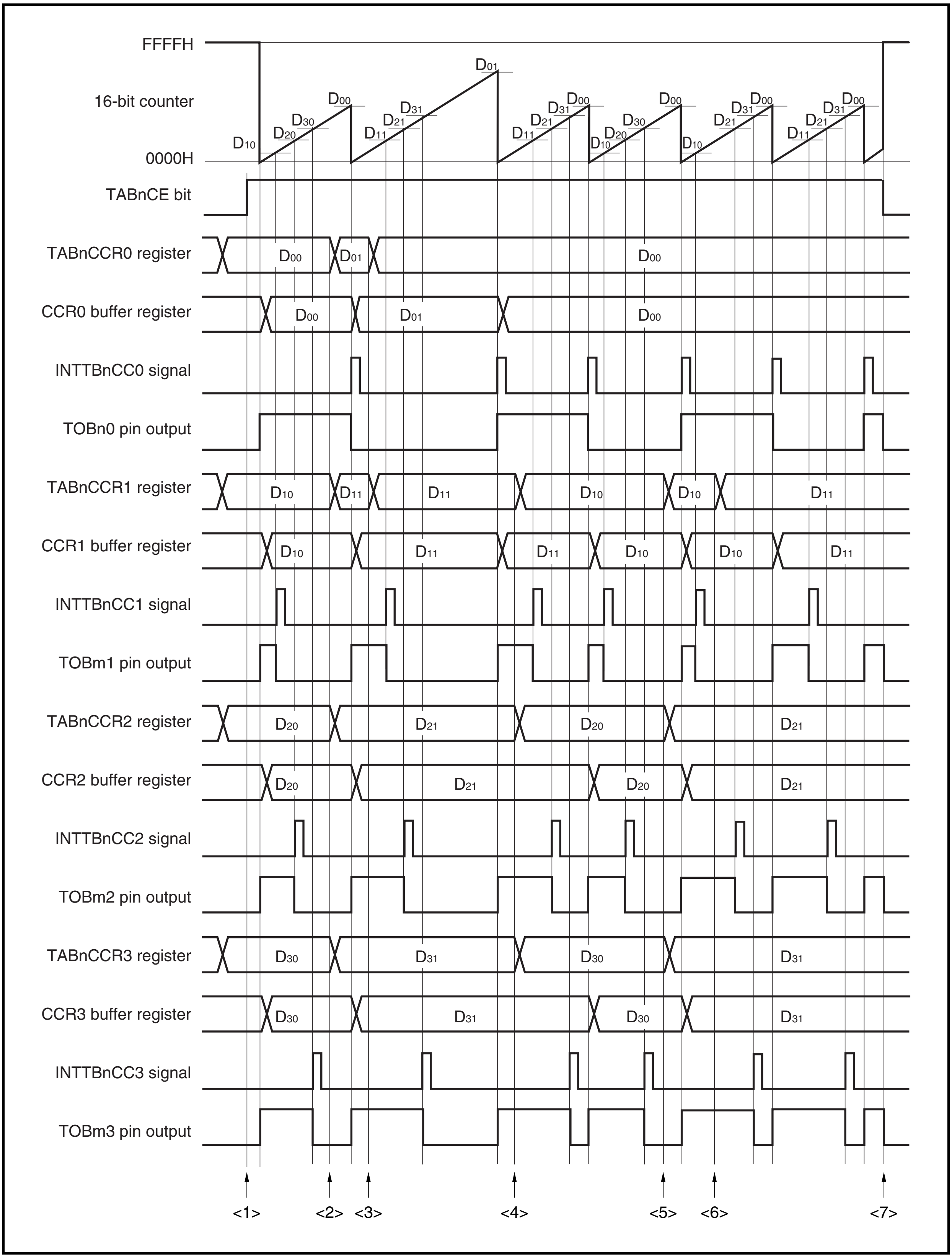
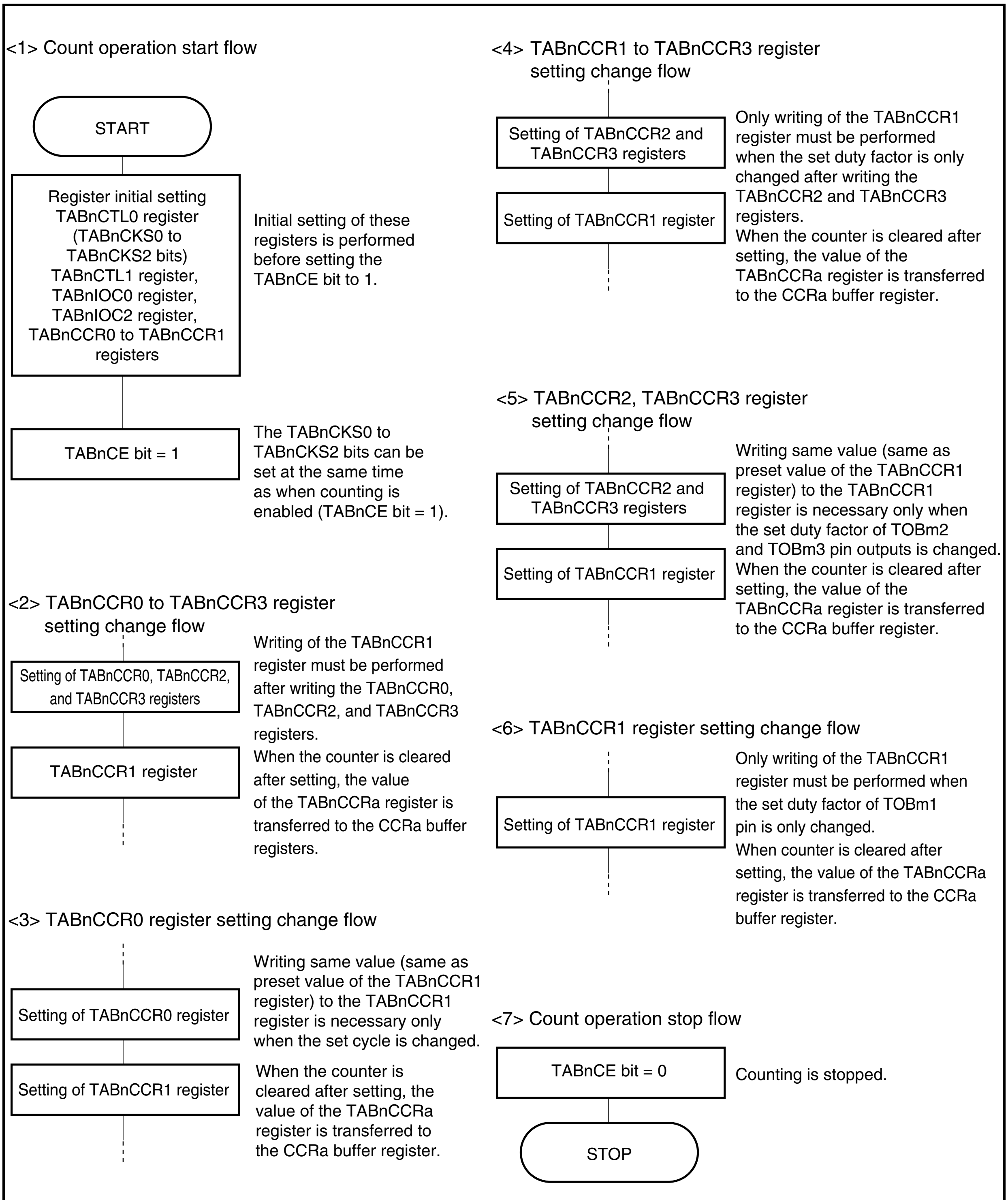


Figure 7-32. Software Processing Flow in PWM Output Mode (2/2)



<1> Count operation start flow

Initial setting of these registers is performed before setting the TABnCE bit to 1.

The TABnCKS0 to TABnCKS2 bits can be set at the same time as when counting is enabled (TABnCE bit = 1).

<2> TABnCCR0 to TABnCCR3 register setting change flow

Writing of the TABnCCR1 register must be performed after writing the TABnCCR0, TABnCCR2, and TABnCCR3 registers.
When the counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer registers.

<3> TABnCCR0 register setting change flow

Writing same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register is necessary only when the set cycle is changed.

When the counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer register.

<4> TABnCCR1 to TABnCCR3 register setting change flow

Setting of TABnCCR2 and TABnCCR3 registers
Setting of TABnCCR1 register

Only writing of the TABnCCR1 register must be performed when the set duty factor is only changed after writing the TABnCCR2 and TABnCCR3 registers.
When the counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer register.

<5> TABnCCR2, TABnCCR3 register setting change flow

Setting of TABnCCR2 and TABnCCR3 registers
Setting of TABnCCR1 register

Writing same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register is necessary only when the set duty factor of TOBm2 and TOBm3 pin outputs is changed.
When the counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer register.

<6> TABnCCR1 register setting change flow

Only writing of the TABnCCR1 register must be performed when the set duty factor of TOBm1 pin is only changed.
When counter is cleared after setting, the value of the TABnCCRa register is transferred to the CCRa buffer register.

<7> Count operation stop flow

Counting is stopped.

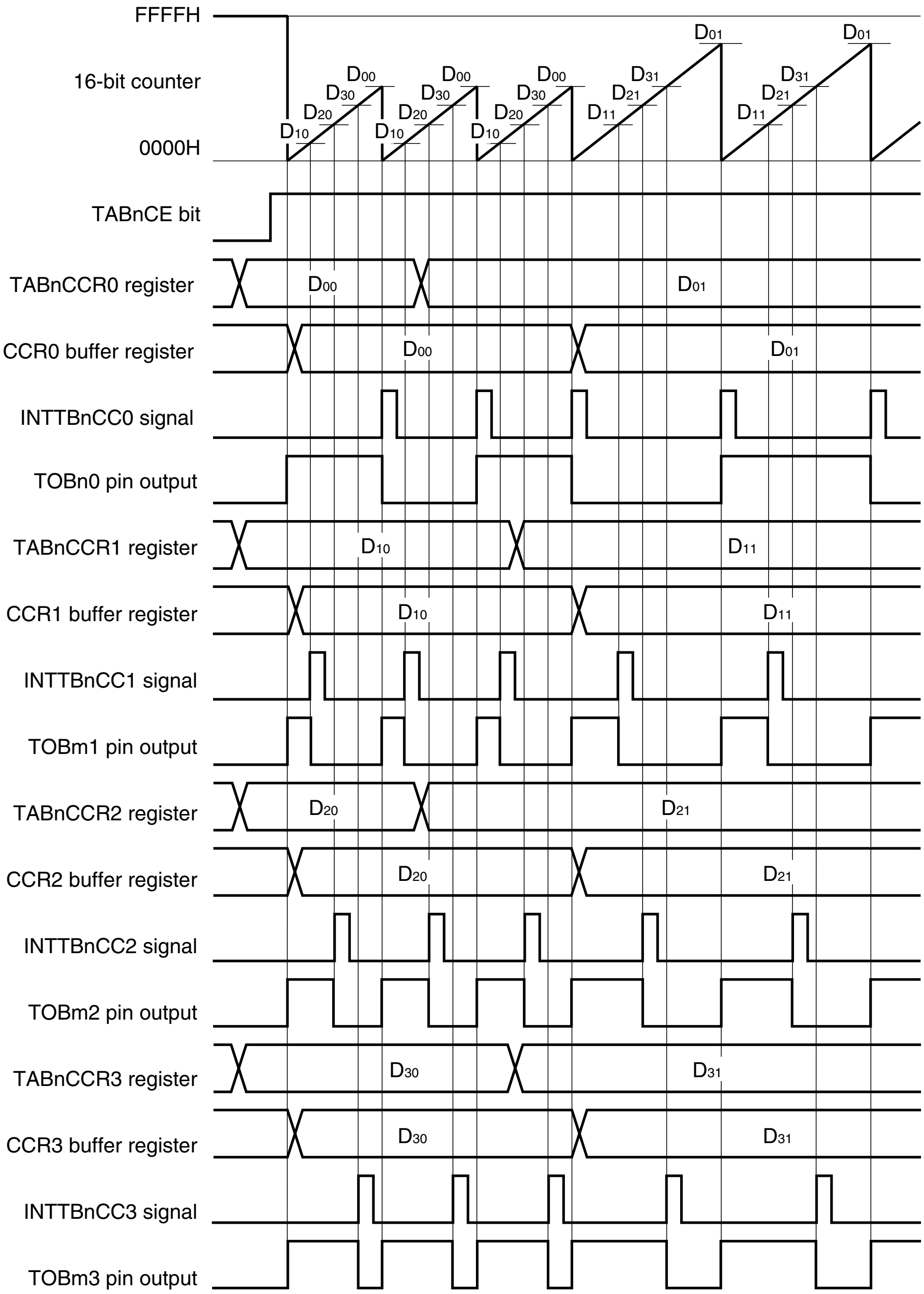
Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TABnCCR1 register last.

Rewrite the TABnCCRa register after writing the TABnCCR1 register after the INTTBnCC0 signal is detected.



Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

To transfer data from the TABnCCRa register to the CCRa buffer register, the TABnCCR1 register must be written.

To change both the cycle and active level of the PWM waveform at this time, first set the cycle to the TABnCCR0 register, set the active level width to the TABnCCR2 and TABnCCR3 registers, and then set an active level width to the TABnCCR1 register.

To change only the cycle of the PWM waveform, first set a cycle to the TABnCCR0 register, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

To change only the active level width (duty factor) of PWM waveform, first set the active level to the TABnCCR2 and TABnCCR3 registers, and then set an active level to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOBm1 pin, only the TABnCCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOBm2 and TOBm3 pins, first set an active level width to the TABnCCR2 and TABnCCR3 registers, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

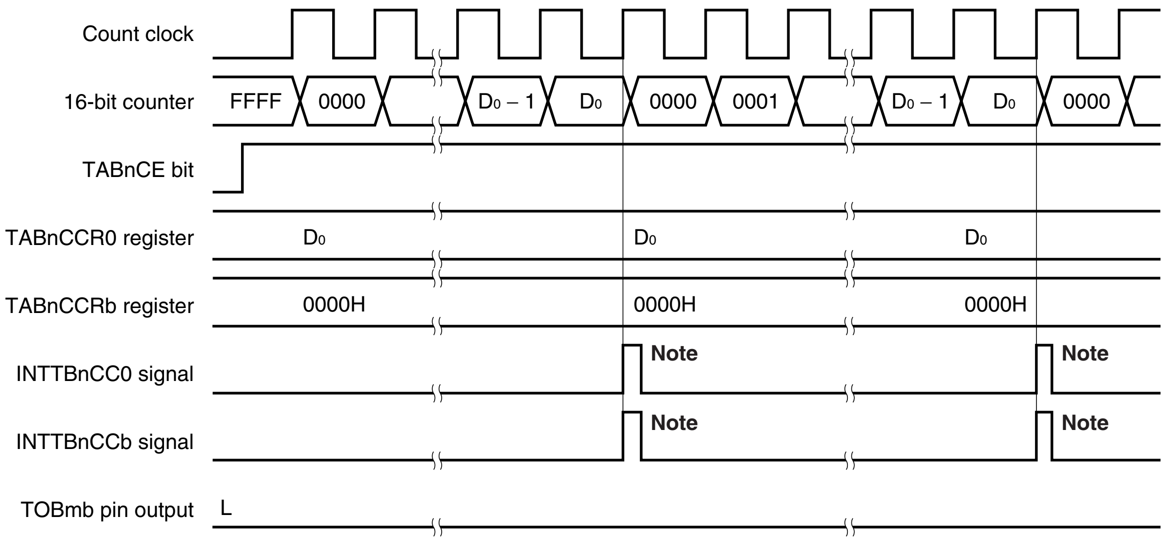
After the TABnCCR1 register is written, the value written to the TABnCCRa register is transferred to the CCRa buffer register in synchronization with the timing of clearing the 16-bit counter, and is used as a value to be compared with the value of the 16-bit counter.

To write the TABnCCR0 to TABnCCR3 registers again after writing the TABnCCR1 register once, do so after the INTTBnCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TABnCCRa register to the CCRa buffer register conflicts with writing the TABnCCRa register.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3

(b) 0%/100% output of PWM waveform

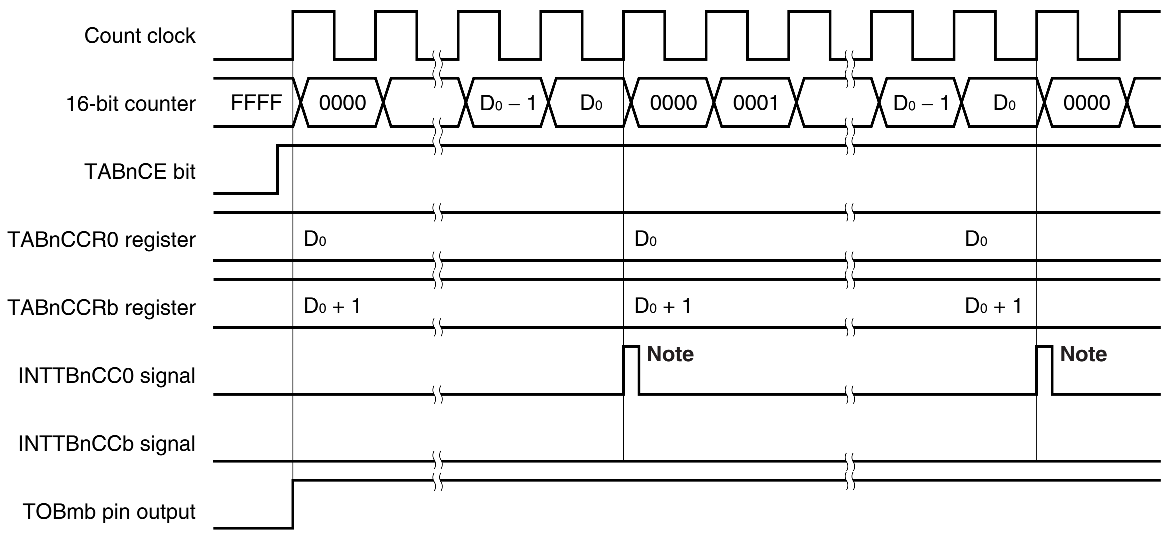
To output a 0% waveform, set the TABnCCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTTBnCC0 and INTTBnCCb signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

To output a 100% waveform, set a value of (set value of TABnCCR0 register + 1) to the TABnCCRb register. If the set value of the TABnCCR0 register is FFFFH, 100% output cannot be produced.

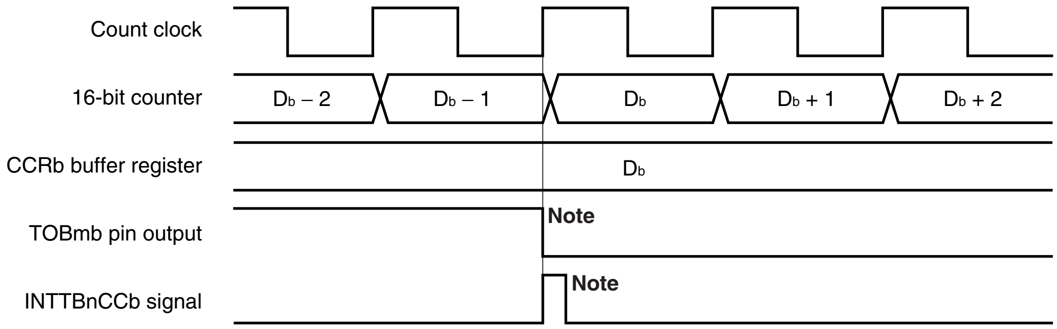


Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

(c) Generation timing of compare match interrupt request signal (INTTBnCCb)

The timing of generation of the INTTBnCCb signal in the PWM output mode differs from the timing of INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the TABnCCRb register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark V850E/IG4-H: n = 0, 1, m = 0, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, b = 1 to 3

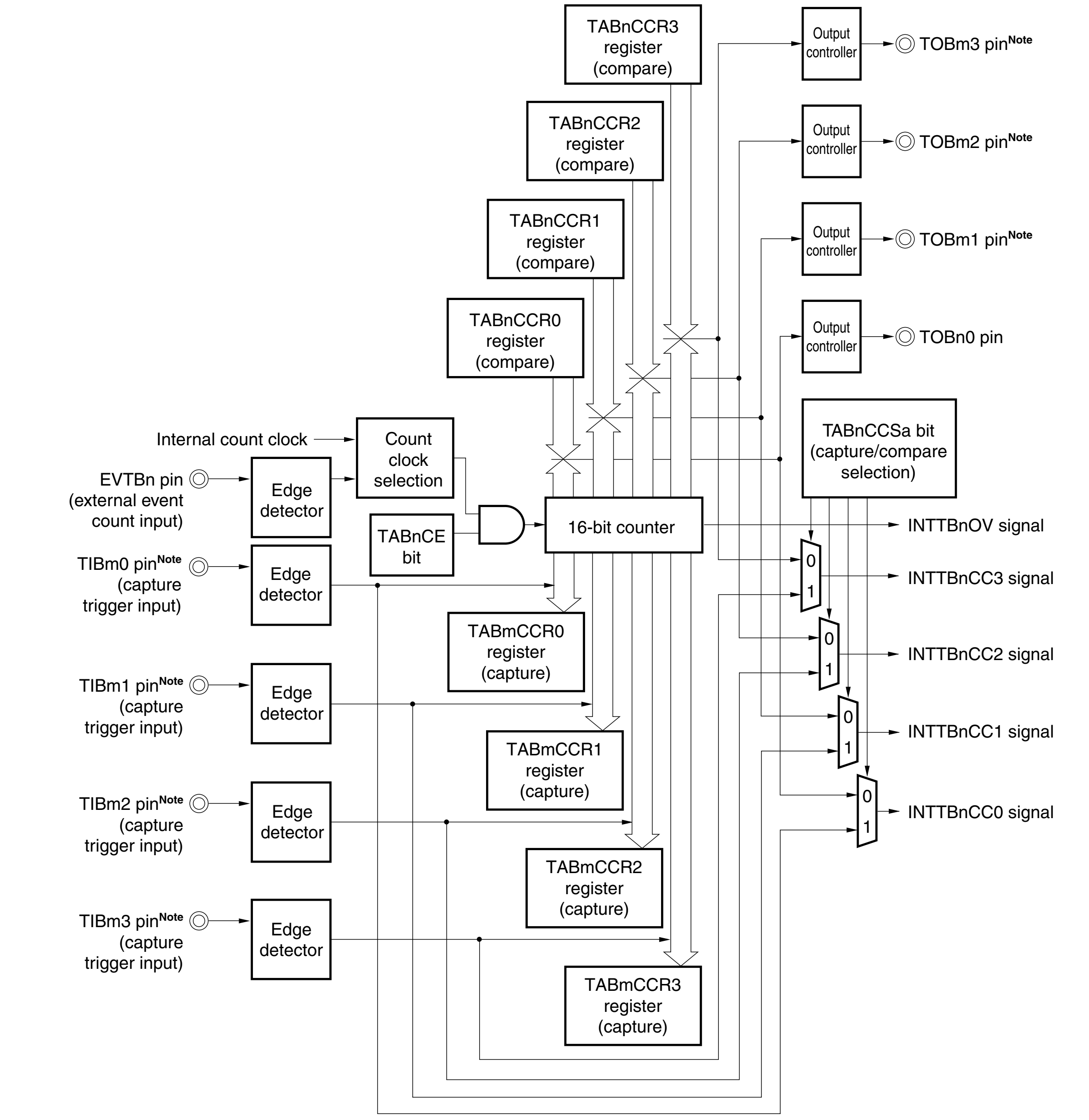
Usually, the INTTBnCCb signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TABnCCRb register.
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOBmb pin.

7.6.6 Free-running timer mode (TABnMD2 to TABnMD0 bits = 101)

In the free-running timer mode, the compare function of TAB0 and TAB1 is valid in both the V850E/IG4-H and V850E/IH4-H. In the V850E/IG4-H, only the capture function of TAB0 is valid. In the V850E/IH4-H, the capture function of both TAB0 and TAB1 is valid.

In the free-running timer mode, 16-bit timer/event counter AB starts counting when the TABnCTL0.TABnCE bit is set to 1. At this time, the TABmCCRa register can be used as a compare register or a capture register, depending on the setting of the TABmOPT0.TABmCCSa bit.

Figure 7-33. Configuration in Free-Running Timer Mode



TABnCCR3 register (compare)
TABnCCR2 register (compare)
TABnCCR1 register (compare)
TABnCCR0 register (compare)
Output controller
TOBm3 pinNote
TOBm2 pinNote
TOBm1 pinNote
TOBn0 pin
TABnCCSa bit (capture/compare selection)
Internal count clock
Count clock selection
EVTBn pin (external event count input)
Edge detector
TABnCE bit
16-bit counter
INTTBnOV signal
INTTBnCC3 signal
INTTBnCC2 signal
INTTBnCC1 signal
INTTBnCC0 signal
0
1
TIBm0 pinNote (capture trigger input)
TIBm1 pinNote (capture trigger input)
TIBm2 pinNote (capture trigger input)
TIBm3 pinNote (capture trigger input)
TABmCCR0 register (capture)
TABmCCR1 register (capture)
TABmCCR2 register (capture)
TABmCCR3 register (capture)

Note Because the capture trigger input pin (TIBmb) and timer output pin (TOBmb) share the same alternate-function pin, the two functions cannot be used at the same time.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3, b = 1 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3, b = 1 to 3

- Compare operation

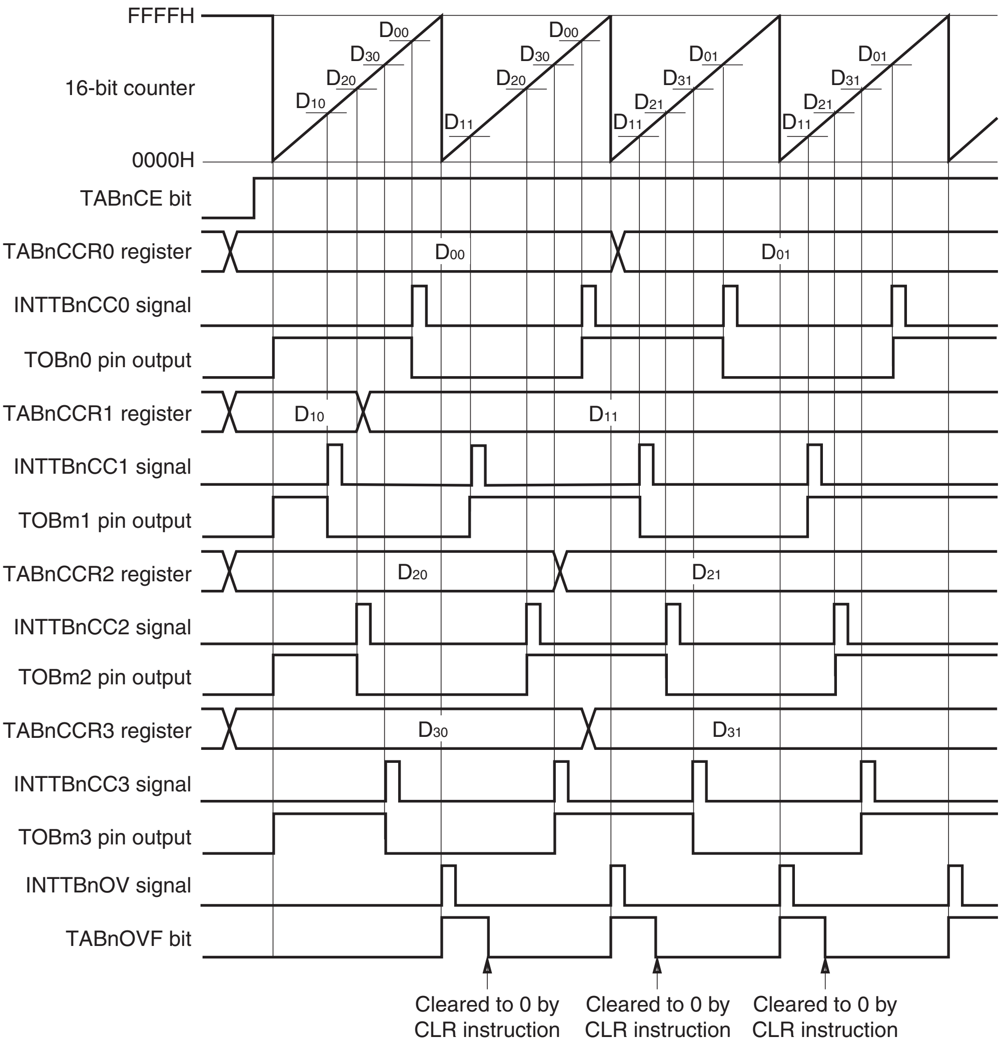
  When the TABnCE bit is set to 1, 16-bit timer/event counter AB starts counting, and the output signals of the TOBn0 and TOBm1 to TOBm3 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TABnCCRa register, a compare match interrupt request signal (INTTBnCCa) is generated, and the output signals of the TOBn0 and TOBm1 to TOBm3 pins are inverted.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTBnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

  The TABnCCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time, and compared with the count value.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3

Figure 7-34. Basic Timing in Free-Running Timer Mode (Compare Function)



Remark V850E/IG4-H: n = 0, 1, m = 0

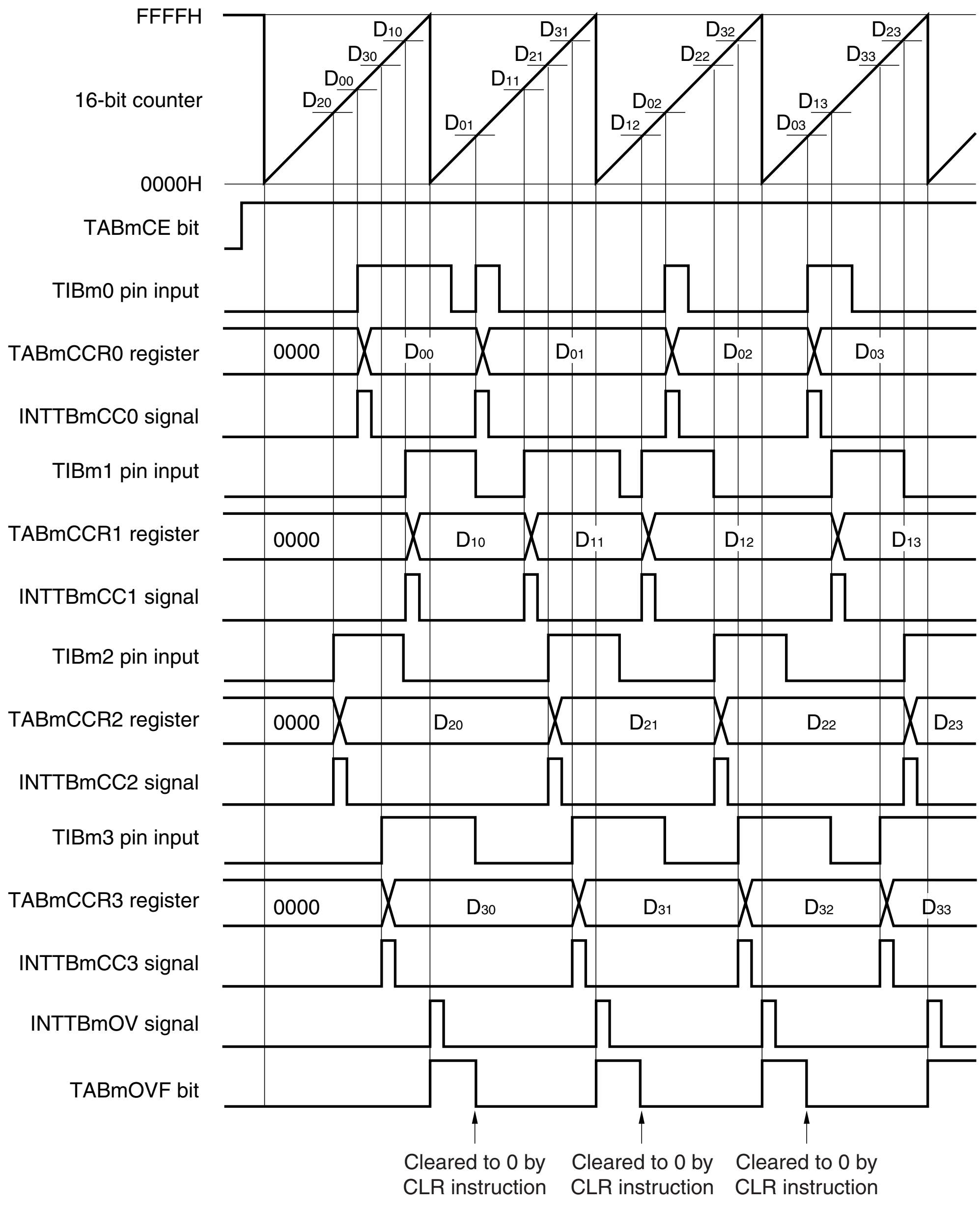
V850E/IH4-H: n = 0, 1, m = 0, 1

- Capture operation

  When the TABmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIBma pin is detected, the count value of the 16-bit counter is stored in the TABmCCRa register, and a capture interrupt request signal (INTTBmCCa) is generated.

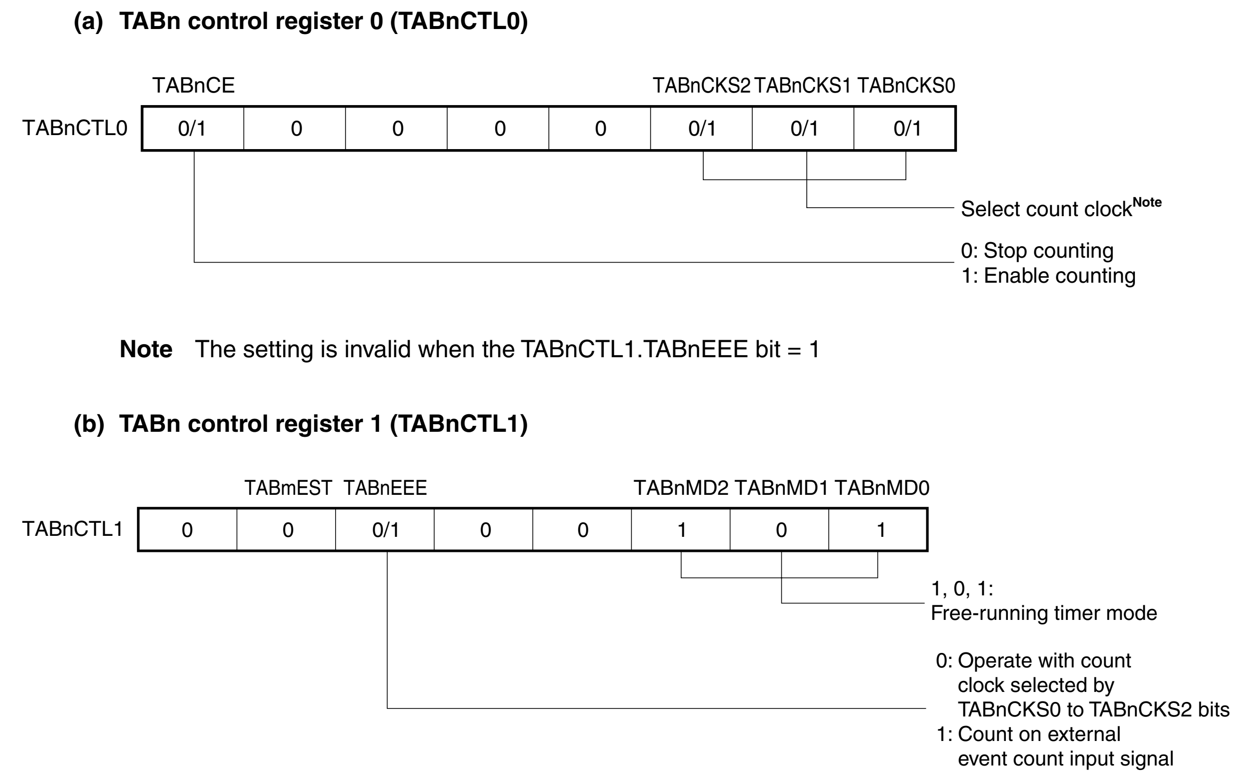
  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTBmOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TABmOPT0.TABmOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

Figure 7-35. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark V850E/IG4-H: m = 0, a = 0 to 3
V850E/IH4-H: m = 0, 1, a = 0 to 3

Figure 7-36. Register Setting in Free-Running Timer Mode (1/3)



(a) TABn control register 0 (TABnCTL0)

	TABnCE					TABnCKS2	TABnCKS1	TABnCKS0
TABnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clockNote
0: Stop counting
1: Enable counting

Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1

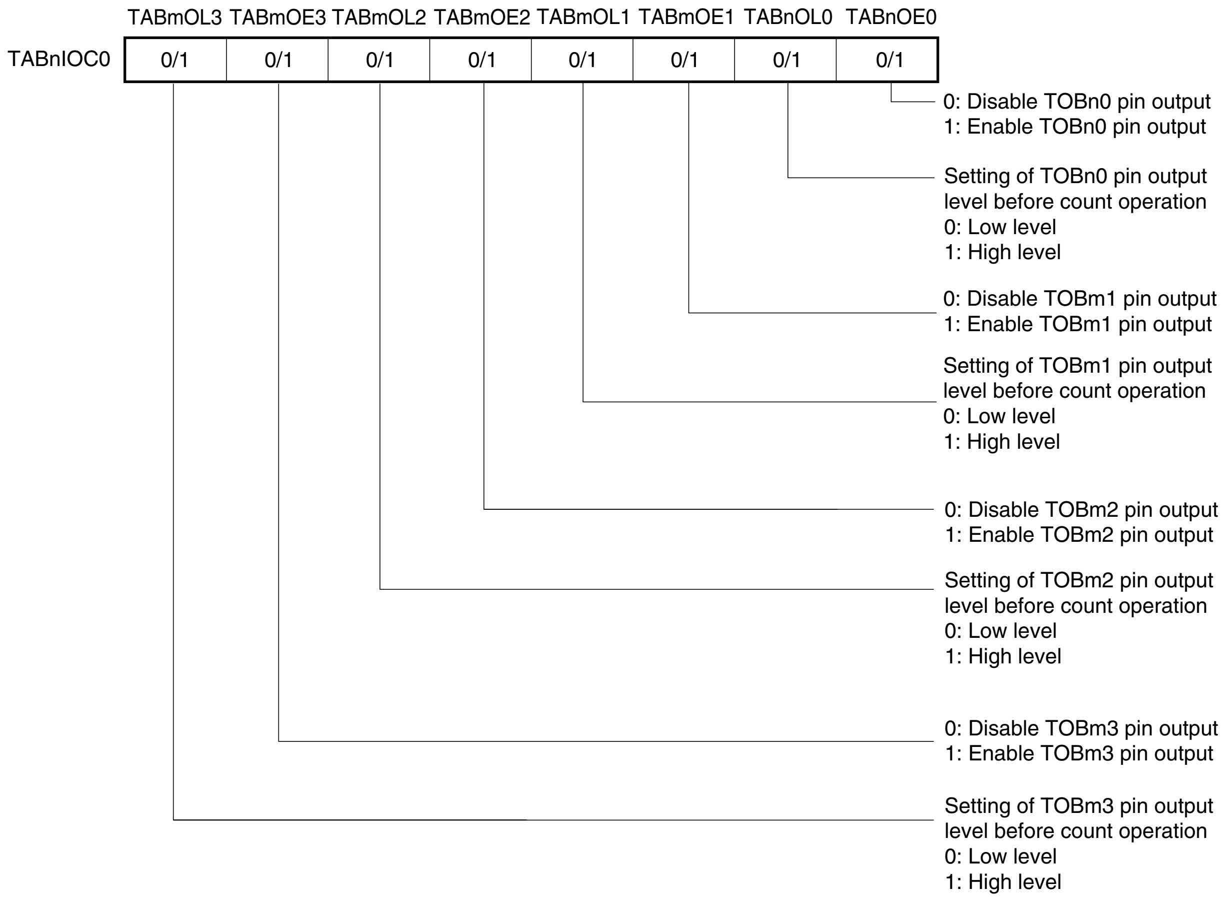
(b) TABn control register 1 (TABnCTL1)

		TABmEST	TABnEEE			TABnMD2	TABnMD1	TABnMD0
TABnCTL1	0	0	0/1	0	0	1	0	1

1, 0, 1:
Free-running timer mode
0: Operate with count clock selected by TABnCKS0 to TABnCKS2 bits
1: Count on external event count input signal

Figure 7-36. Register Setting in Free-Running Timer Mode (2/3)

(c) TABn I/O control register 0 (TABnIOC0)



	TABmOL3	TABmOE3	TABmOL2	TABmOE2	TABmOL1	TABmOE1	TABnOL0	TABnOE0
TABnIOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

0: Disable TOBn0 pin output
1: Enable TOBn0 pin output

Setting of TOBn0 pin output level before count operation
0: Low level
1: High level

0: Disable TOBm1 pin output
1: Enable TOBm1 pin output

Setting of TOBm1 pin output level before count operation
0: Low level
1: High level

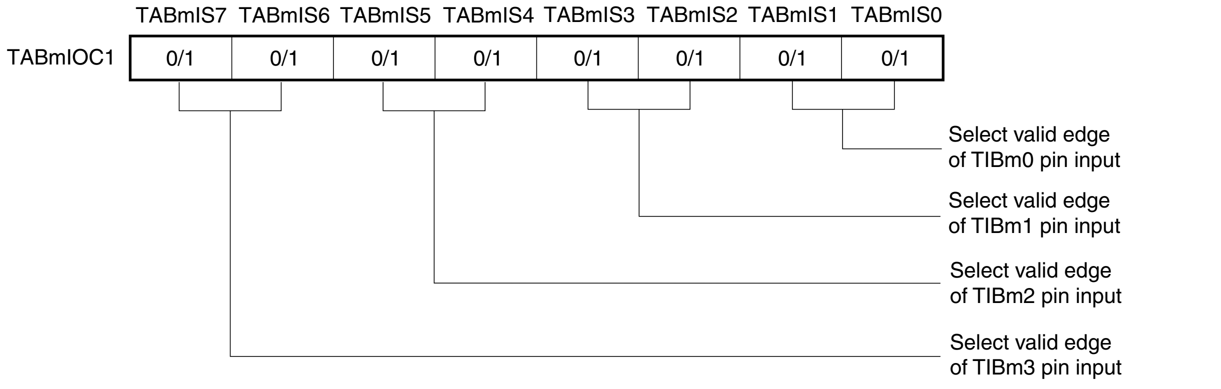
0: Disable TOBm2 pin output
1: Enable TOBm2 pin output

Setting of TOBm2 pin output level before count operation
0: Low level
1: High level

0: Disable TOBm3 pin output
1: Enable TOBm3 pin output

Setting of TOBm3 pin output level before count operation
0: Low level
1: High level

(d) TABm I/O control register 1 (TABmIOC1)



	TABmIS7	TABmIS6	TABmIS5	TABmIS4	TABmIS3	TABmIS2	TABmIS1	TABmIS0
TABmIOC1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Select valid edge of TIBm0 pin input

Select valid edge of TIBm1 pin input

Select valid edge of TIBm2 pin input

Select valid edge of TIBm3 pin input

Figure 7-36. Register Setting in Free-Running Timer Mode (3/3)

(e) TABn I/O control register 2 (TABnIOC2)

					TABnEES1	TABnEES0	TABmETS1	TABmETS0
TABnIOC2	0	0	0	0	0/1	0/1	0	0

TABnEES1, TABnEES0: Select valid edge of external event count input (EVTBn pin)

(f) TABn option register 0 (TABnOPT0)

	TABmCCS3	TABmCCS2	TABmCCS1	TABmCCS0		TABnCMS	TABnCUF	TABnOVF
TABnOPT0	0/1	0/1	0/1	0/1	0	0	0	0/1

TABnOVF: Overflow flag

TABmCCS0: Specifies if TABmCCR0 register functions as capture or compare register
0: Compare register
1: Capture register

TABmCCS1: Specifies if TABmCCR1 register functions as capture or compare register
0: Compare register
1: Capture register

TABmCCS2: Specifies if TABmCCR2 register functions as capture or compare register
0: Compare register
1: Capture register

TABmCCS3: Specifies if TABmCCR3 register functions as capture or compare register
0: Compare register
1: Capture register

(g) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

(h) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

These registers function as capture registers or compare registers depending on the setting of the TABmOPT0.TABmCCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIBma pin is detected.

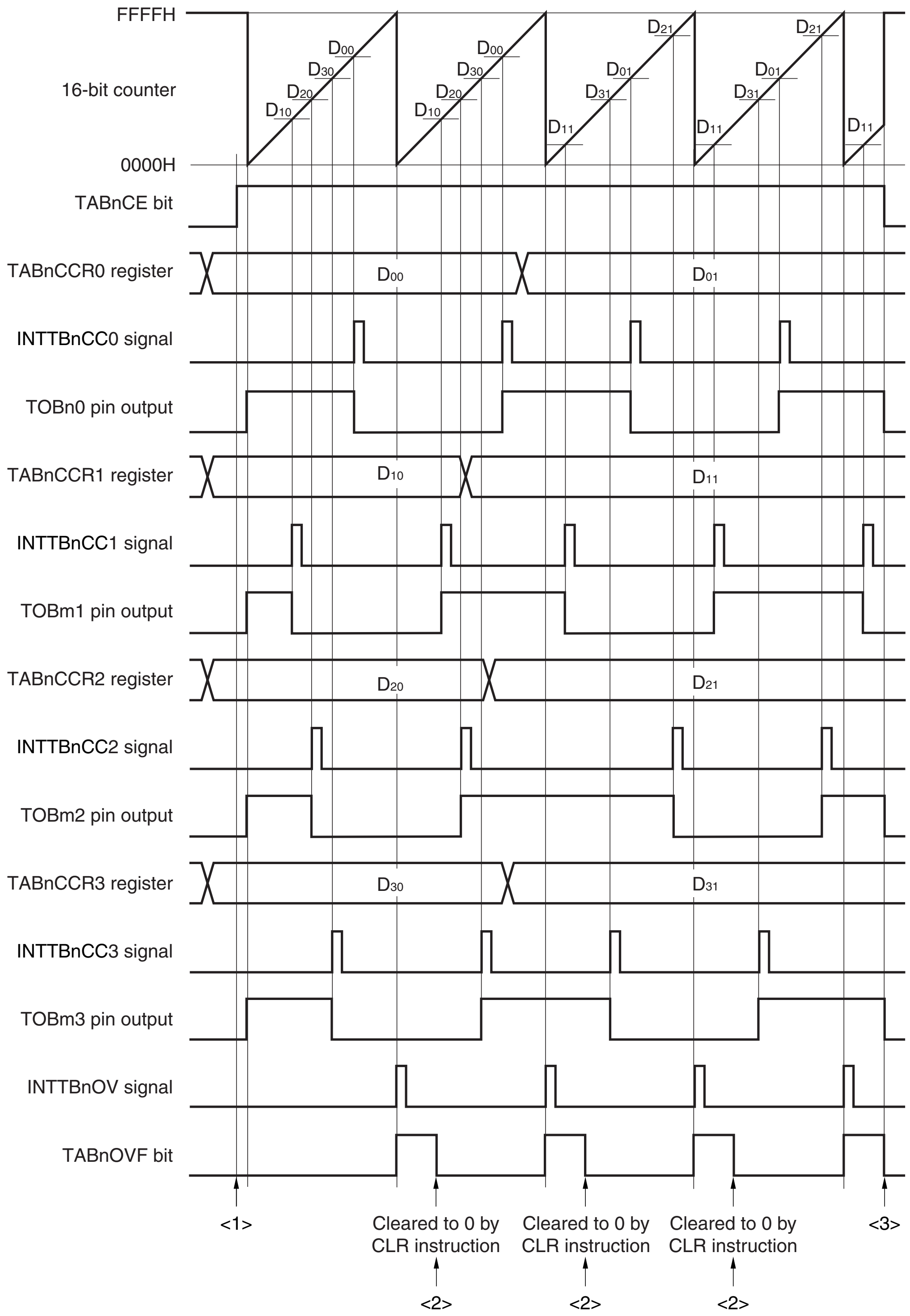
When the registers function as compare registers and when D_a is set to the TABnCCRa register, the INTTBnCCa signal is generated when the counter reaches (D_a + 1), and the output signals of the TOBn0 and TOBm1 to TOBm3 pins are inverted.

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3

(1) Operation flow in free-running timer mode

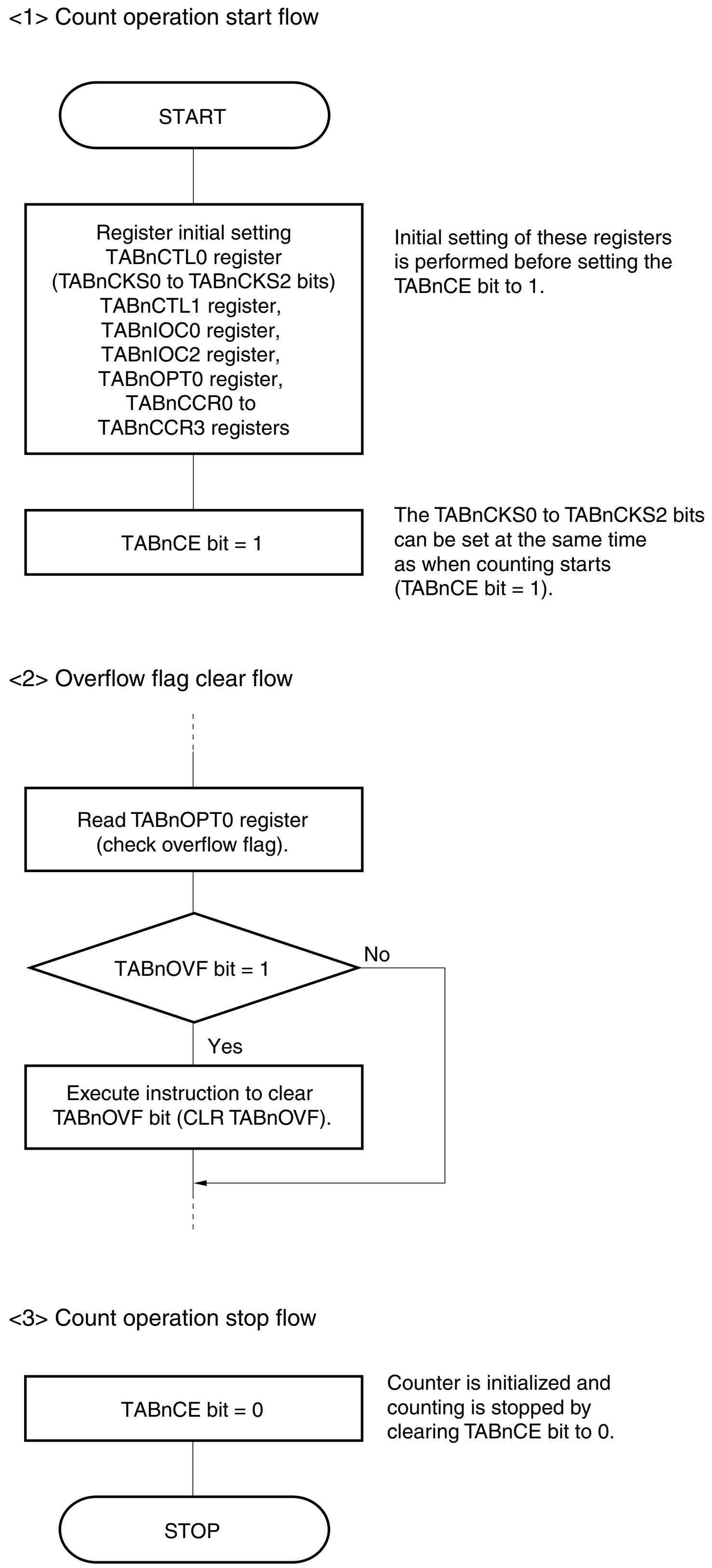
(a) When using capture/compare register as compare register

Figure 7-37. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



Remark V850E/IG4-H: n = 0, 1, m = 0
V850E/IH4-H: n = 0, 1, m = 0, 1

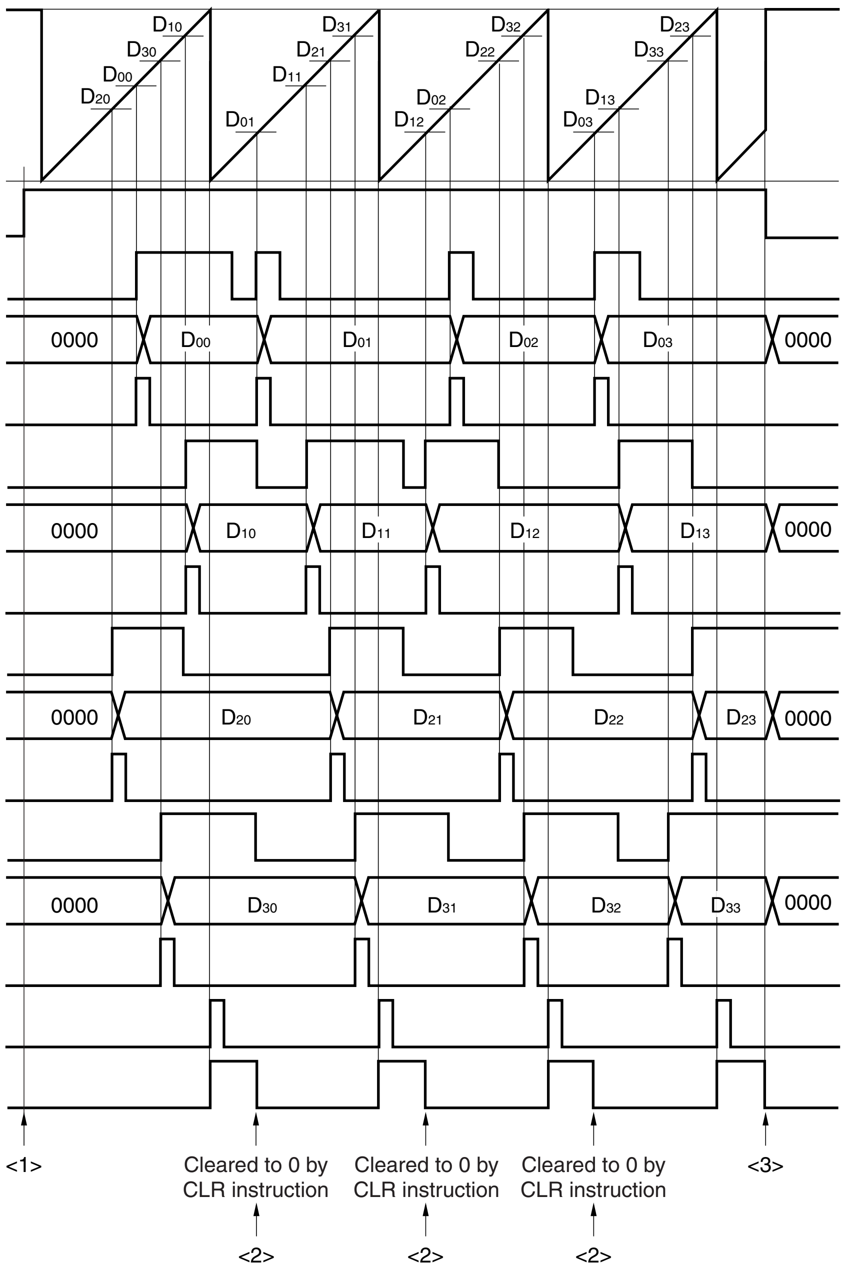
Figure 7-37. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)


<1> Count operation start flow
START
Register initial setting TABnCTL0 register (TABnCKS0 to TABnCKS2 bits) TABnCTL1 register, TABnIOC0 register, TABnIOC2 register, TABnOPT0 register, TABnCCR0 to TABnCCR3 registers
Initial setting of these registers is performed before setting the TABnCE bit to 1.
TABnCE bit = 1
The TABnCKS0 to TABnCKS2 bits can be set at the same time as when counting starts (TABnCE bit = 1).
<2> Overflow flag clear flow
Read TABnOPT0 register (check overflow flag).
TABnOVF bit = 1
No
Yes
Execute instruction to clear TABnOVF bit (CLR TABnOVF).
<3> Count operation stop flow
TABnCE bit = 0
Counter is initialized and counting is stopped by clearing TABnCE bit to 0.
STOP

Remark n = 0, 1

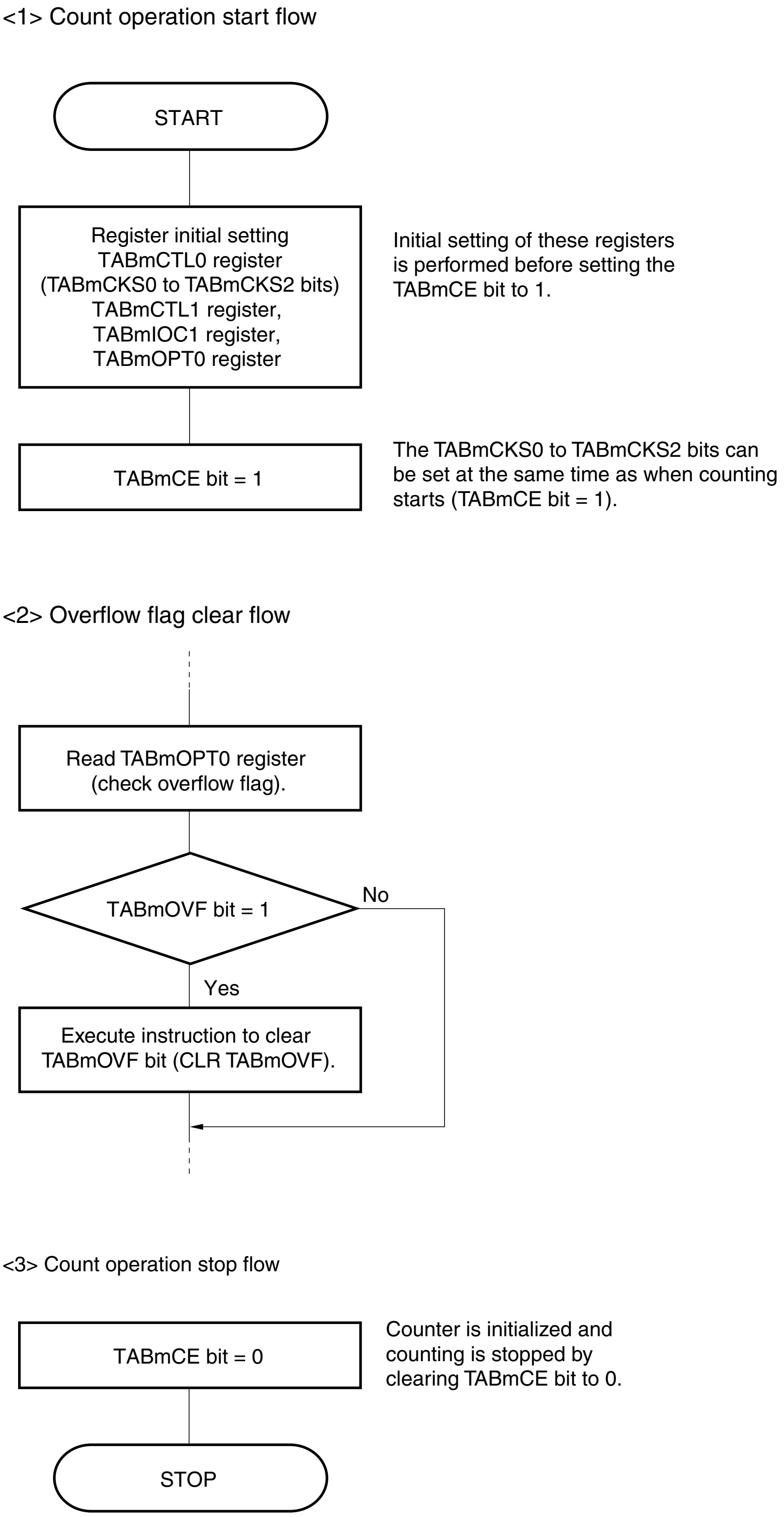
(b) When using capture/compare register as capture register

Figure 7-38. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)



Remark V850E/IG4-H: m = 0
V850E/IH4-H: m = 0, 1

Figure 7-38. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



<1> Count operation start flow

Initial setting of these registers is performed before setting the TABmCE bit to 1.

The TABmCKS0 to TABmCKS2 bits can be set at the same time as when counting starts (TABmCE bit = 1).

<2> Overflow flag clear flow

<3> Count operation stop flow

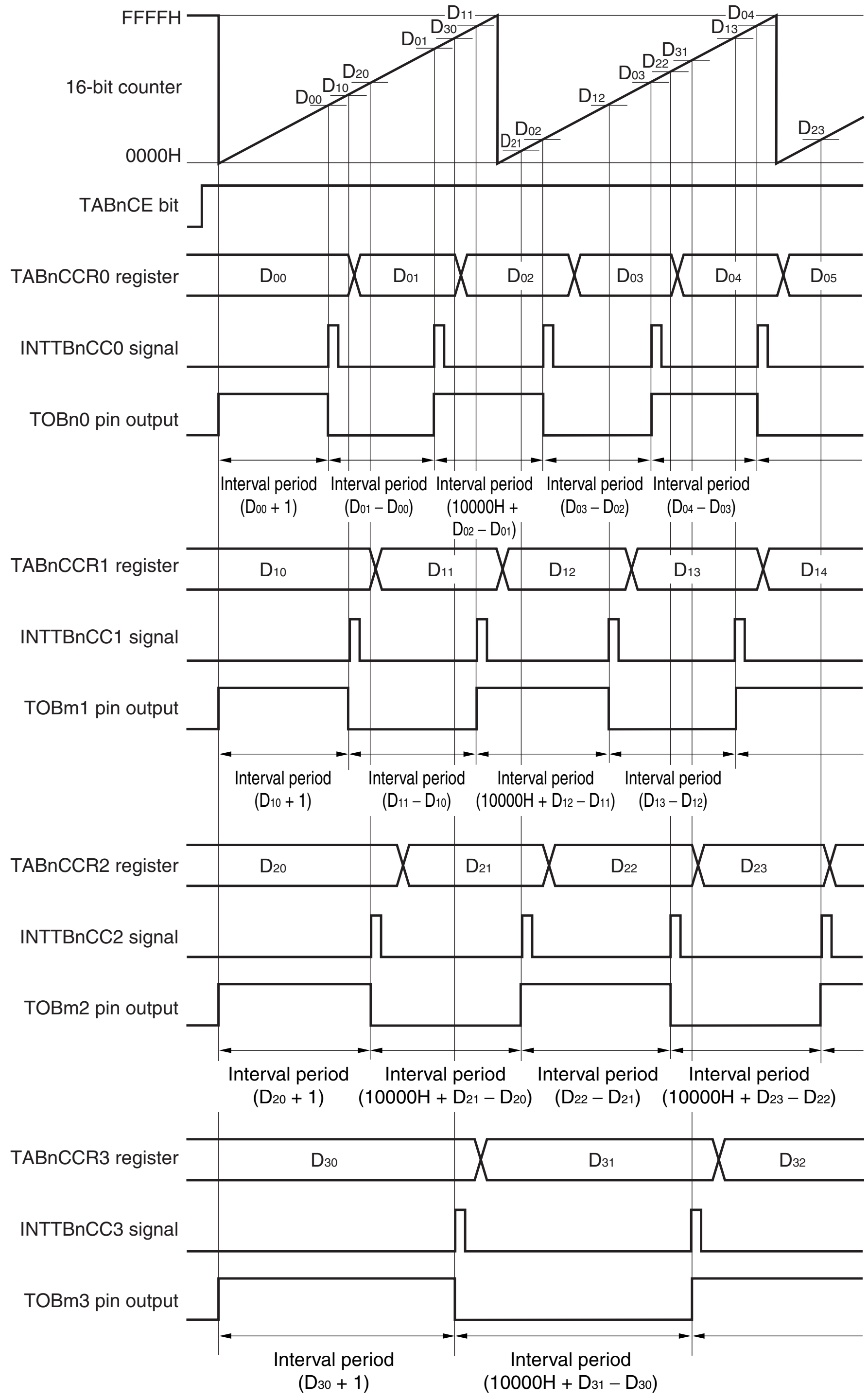
Counter is initialized and counting is stopped by clearing TABmCE bit to 0.

Remark V850E/IG4-H: m = 0
V850E/IH4-H: m = 0, 1

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter AB is used as an interval timer with the TABnCCRa register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTBnCCa signal has been detected.


FFFFH
16-bit counter
0000H
D00
D10
D20
D01
D30
D11
D21
D02
D12
D03
D22
D31
D13
D04
D23
TABnCE bit
TABnCCR0 register
D00
D01
D02
D03
D04
D05
INTTBnCC0 signal
TOBn0 pin output
Interval period (D00 + 1)
Interval period (D01 – D00)
Interval period (10000H + D02 – D01)
Interval period (D03 – D02)
Interval period (D04 – D03)
TABnCCR1 register
D10
D11
D12
D13
D14
INTTBnCC1 signal
TOBm1 pin output
Interval period (D10 + 1)
Interval period (D11 – D10)
Interval period (10000H + D12 – D11)
Interval period (D13 – D12)
TABnCCR2 register
D20
D21
D22
D23
INTTBnCC2 signal
TOBm2 pin output
Interval period (D20 + 1)
Interval period (10000H + D21 – D20)
Interval period (D22 – D21)
Interval period (10000H + D23 – D22)
TABnCCR3 register
D30
D31
D32
INTTBnCC3 signal
TOBm3 pin output
Interval period (D30 + 1)
Interval period (10000H + D31 – D30)

Remark V850E/IG4-H: n = 0, 1, m = 0, a = 0 to 3
V850E/IH4-H: n = 0, 1, m = 0, 1, a = 0 to 3

When performing an interval operation in the free-running timer mode, four intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TABnCCRa register must be re-set in the interrupt servicing that is executed when the INTTBnCCa signal is detected.

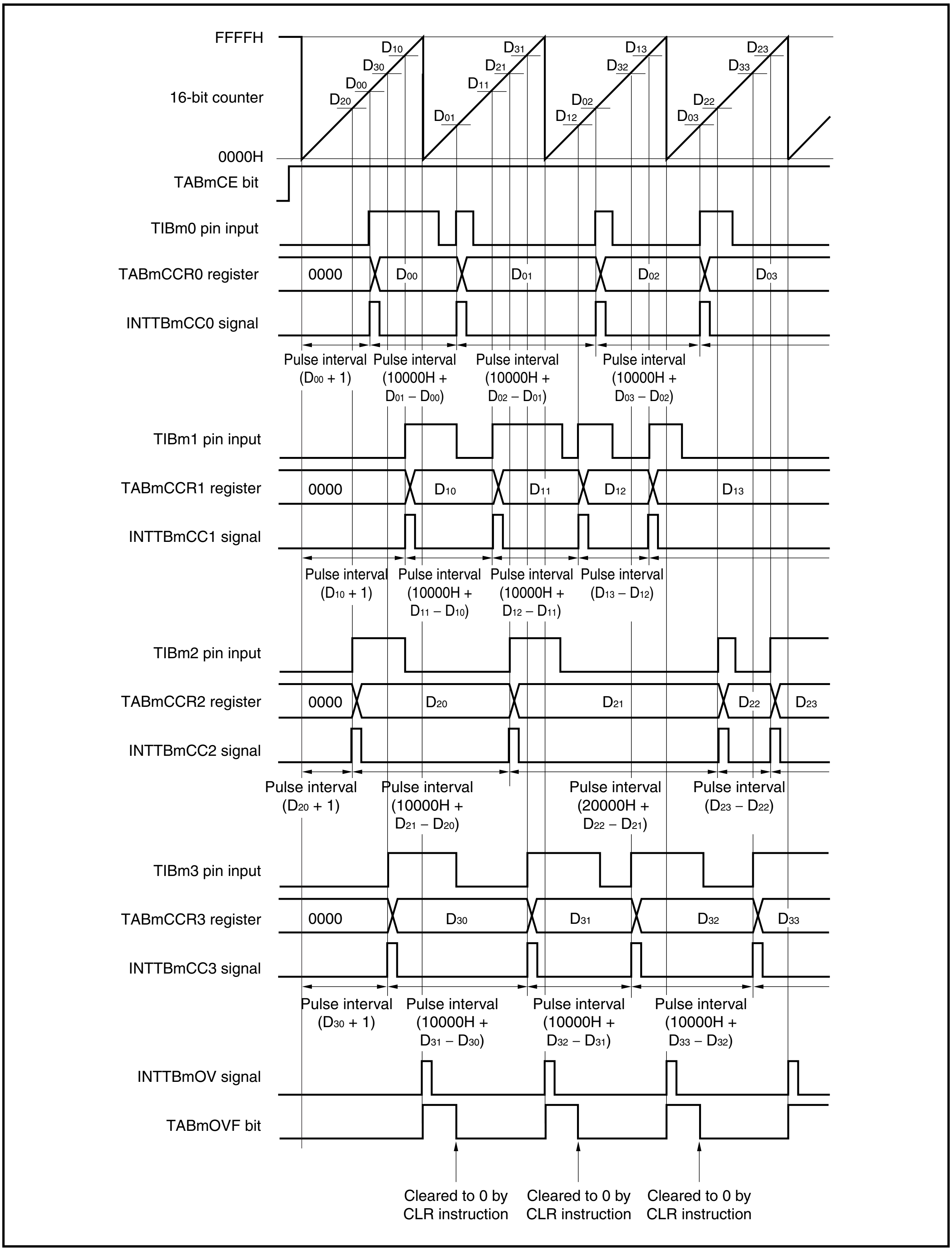
The set value for re-setting the TABnCCRa register can be calculated by the following expression, where "D_a" is the interval period.

Compare register default value: $D_a - 1$
Value set to compare register second and subsequent time: Previous set value + D_a
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0, 1
a = 0 to 3

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TABmCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTBmCCa signal has been detected and for calculating an interval.


FFFFH
16-bit counter
0000H
D10
D30
D00
D20
D31
D21
D11
D01
D13
D32
D02
D12
D23
D33
D22
D03
TABmCE bit
TIBm0 pin input
TABmCCR0 register
0000
D00
D01
D02
D03
INTTBmCC0 signal
Pulse interval (D00 + 1)
Pulse interval (10000H + D01 – D00)
Pulse interval (10000H + D02 – D01)
Pulse interval (10000H + D03 – D02)
TIBm1 pin input
TABmCCR1 register
0000
D10
D11
D12
D13
INTTBmCC1 signal
Pulse interval (D10 + 1)
Pulse interval (10000H + D11 – D10)
Pulse interval (10000H + D12 – D11)
Pulse interval (D13 – D12)
TIBm2 pin input
TABmCCR2 register
0000
D20
D21
D22
D23
INTTBmCC2 signal
Pulse interval (D20 + 1)
Pulse interval (10000H + D21 – D20)
Pulse interval (20000H + D22 – D21)
Pulse interval (D23 – D22)
TIBm3 pin input
TABmCCR3 register
0000
D30
D31
D32
D33
INTTBmCC3 signal
Pulse interval (D30 + 1)
Pulse interval (10000H + D31 – D30)
Pulse interval (10000H + D32 – D31)
Pulse interval (10000H + D33 – D32)
INTTBmOV signal
TABmOVF bit
Cleared to 0 by CLR instruction
Cleared to 0 by CLR instruction
Cleared to 0 by CLR instruction

When executing pulse width measurement in the free-running timer mode, four pulse widths can be measured with one channel.

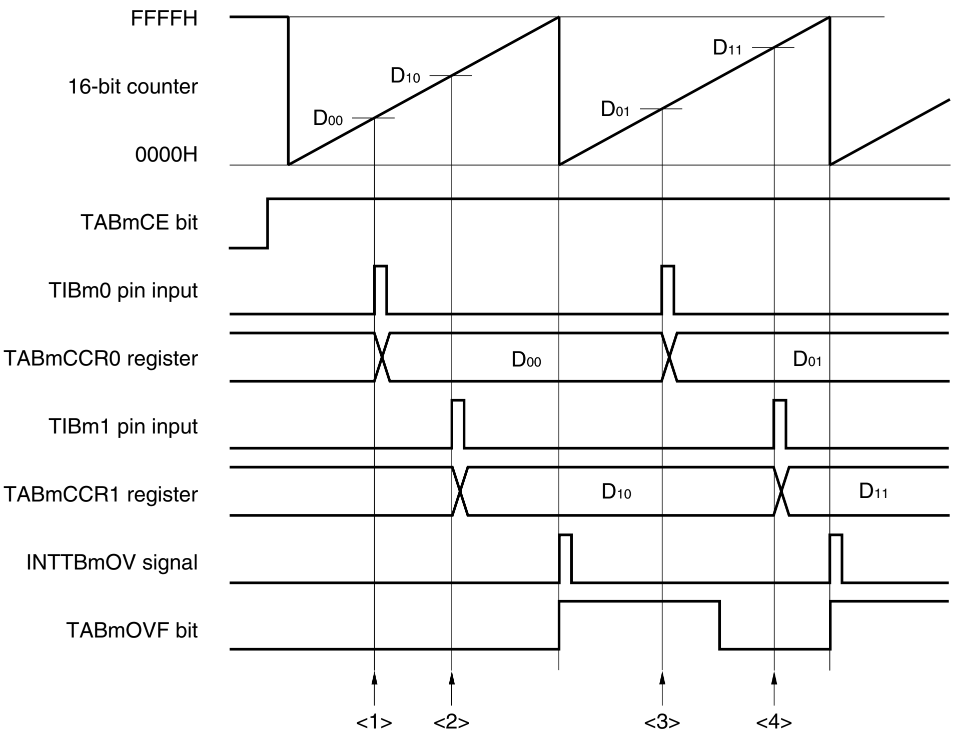
To measure a pulse width, the pulse width can be calculated by reading the value of the TABmCCRa register in synchronization with the INTTBmCCa signal, and calculating the difference between the read value and the previously read value.

Remark V850E/IG4-H: m = 0, a = 0 to 3
V850E/IH4-H: m = 0, 1, a = 0 to 3

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TABmCCR0 register (setting of the default value of the TIBm0 pin input).
<2> Read the TABmCCR1 register (setting of the default value of the TIBm1 pin input).
<3> Read the TABmCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<4> Read the TABmCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} − D_{10}) (incorrect).

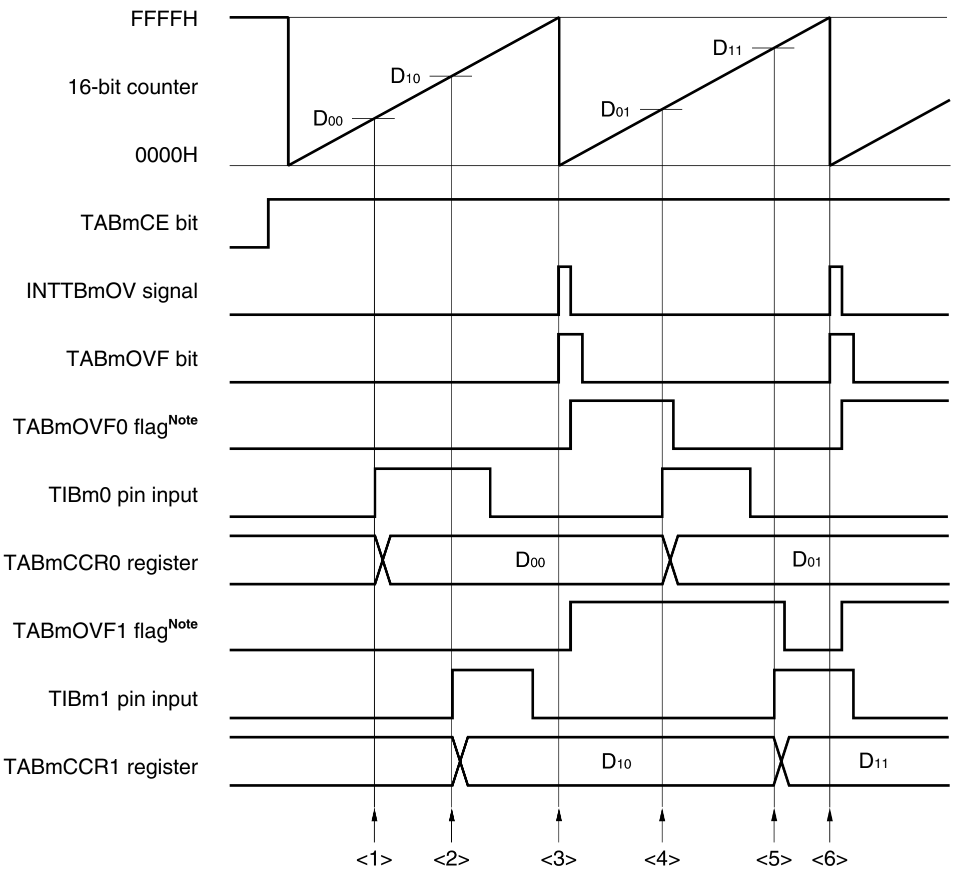
Remark V850E/IG4-H: m = 0
V850E/IH4-H: m = 0, 1

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.

Use software when using two capture registers. An example of how to use software is shown below.

(1/2)

Example when two capture registers are used (using overflow interrupt)



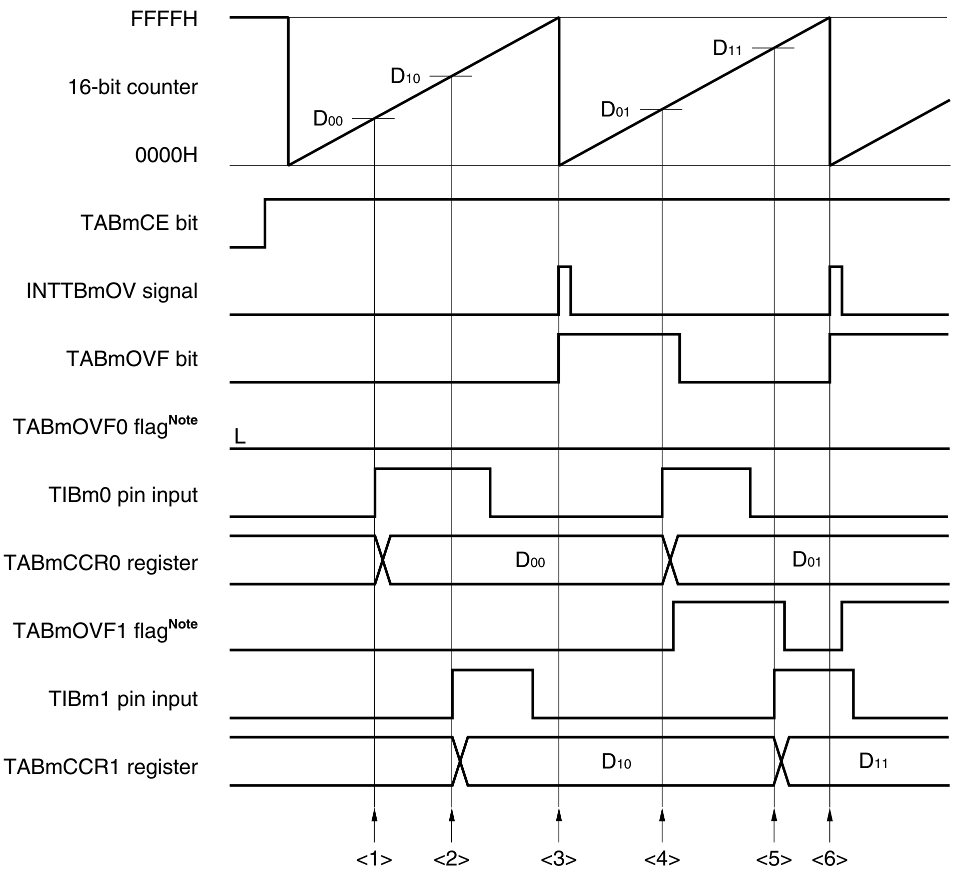
Note The TABmOVF0 and TABmOVF1 flags are set on the internal RAM by software.

<1> Read the TABmCCR0 register (setting of the default value of the TIBm0 pin input).
<2> Read the TABmCCR1 register (setting of the default value of the TIBm1 pin input).
<3> An overflow occurs. Set the TABmOVF0 and TABmOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.
<4> Read the TABmCCR0 register.
Read the TABmOVF0 flag. If the TABmOVF0 flag is 1, clear it to 0.
Because the TABmOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<5> Read the TABmCCR1 register.
Read the TABmOVF1 flag. If the TABmOVF1 flag is 1, clear it to 0 (the TABmOVF0 flag is cleared in <4>, and the TABmOVF1 flag remains 1).
Because the TABmOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).
<6> Same as <3>

Remark V850E/IG4-H: m = 0
V850E/IH4-H: m = 0, 1

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TABmOVF0 and TABmOVF1 flags are set on the internal RAM by software.

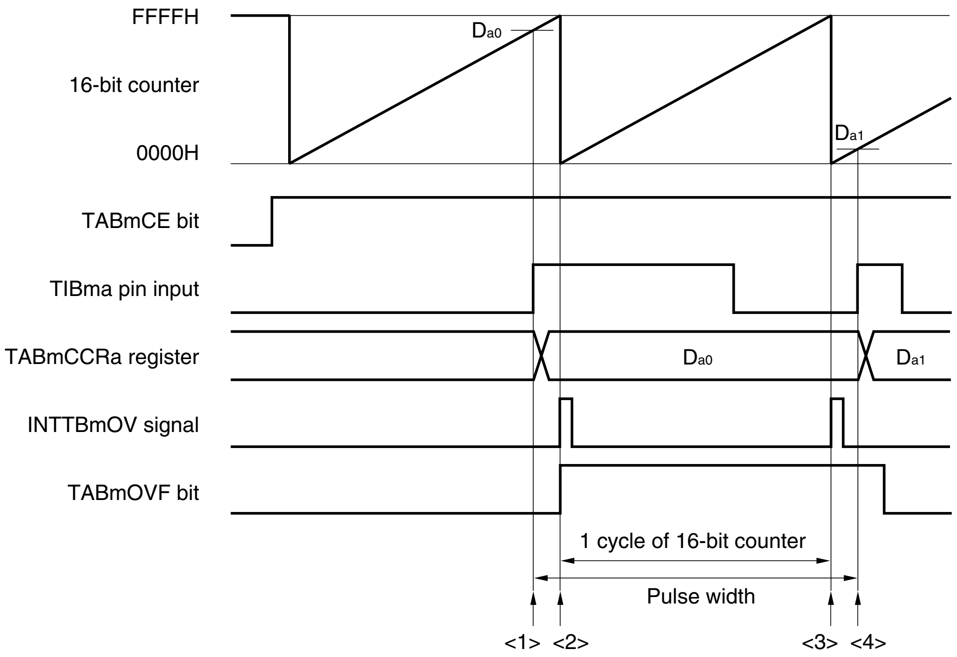
<1> Read the TABmCCR0 register (setting of the default value of the TIBm0 pin input).
<2> Read the TABmCCR1 register (setting of the default value of the TIBm1 pin input).
<3> An overflow occurs. Nothing is done by software.
<4> Read the TABmCCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TABmOVF1 flag to 1, and clear the overflow flag to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<5> Read the TABmCCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TABmOVF1 flag. If the TABmOVF1 flag is 1, clear it to 0.
Because the TABmOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).
<6> Same as <3>

Remark V850E/IG4-H: m = 0
V850E/IH4-H: m = 0, 1

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



The following problem may occur when a long pulse width in the free-running timer mode.

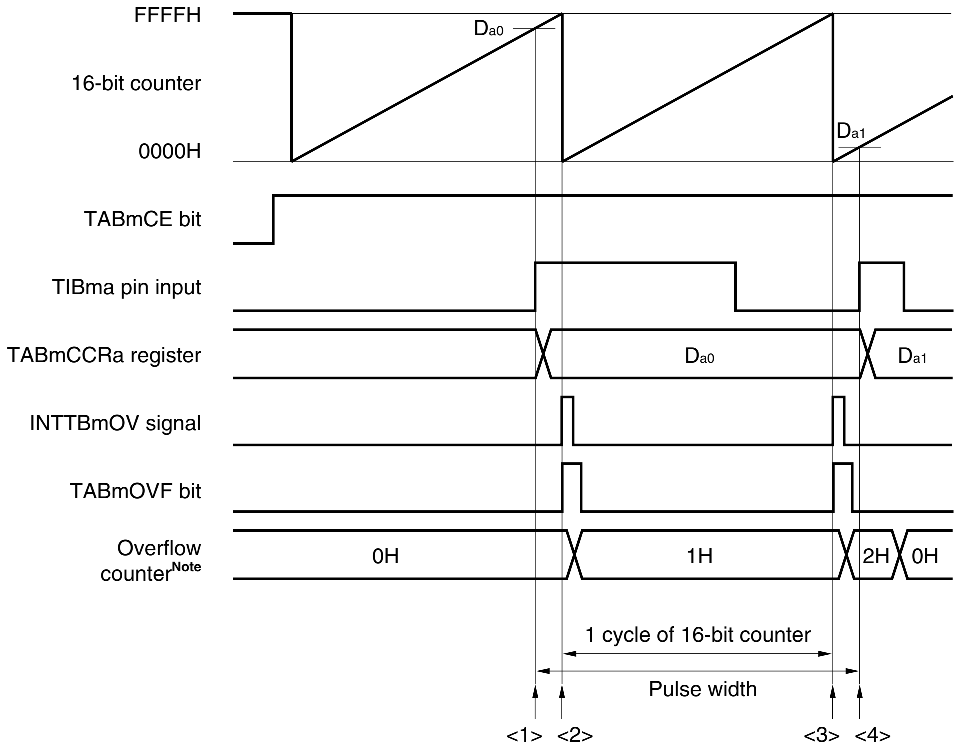
<1> Read the TABmCCRa register (setting of the default value of the TIBma pin input).
<2> An overflow occurs. Nothing is done by software.
<3> An overflow occurs a second time. Nothing is done by software.
<4> Read the TABmCCRa register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{a1} − D_{a0}) (incorrect).
Actually, the pulse width must be (20000H + D_{a1} − D_{a0}) because an overflow occurs twice.

Remark V850E/IG4-H: m = 0
V850E/IH4-H: m = 0, 1

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.

If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TABmCCRa register (setting of the default value of the TIBma pin input).

<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<3> An overflow occurs a second time. Increment (+1) the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<4> Read the TABmCCRa register.
Read the overflow counter.
→ When the overflow counter is “N”, the pulse width can be calculated by (N × 10000H + D_{a1} – D_{a0}).
In this example, the pulse width is (20000H + D_{a1} – D_{a0}) because an overflow occurs twice.
Clear the overflow counter (0H).

Remark V850E/IG4-H: m = 0, a = 0 to 3
V850E/IH4-H: m = 0, 1, a = 0 to 3

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TABmOVF bit to 0 with the CLR instruction after reading the TABmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TABmOPT0 register after reading the TABmOVF bit when it is 1.

7.6.7 Pulse width measurement mode (TABmMD2 to TABmMD0 bits = 110)

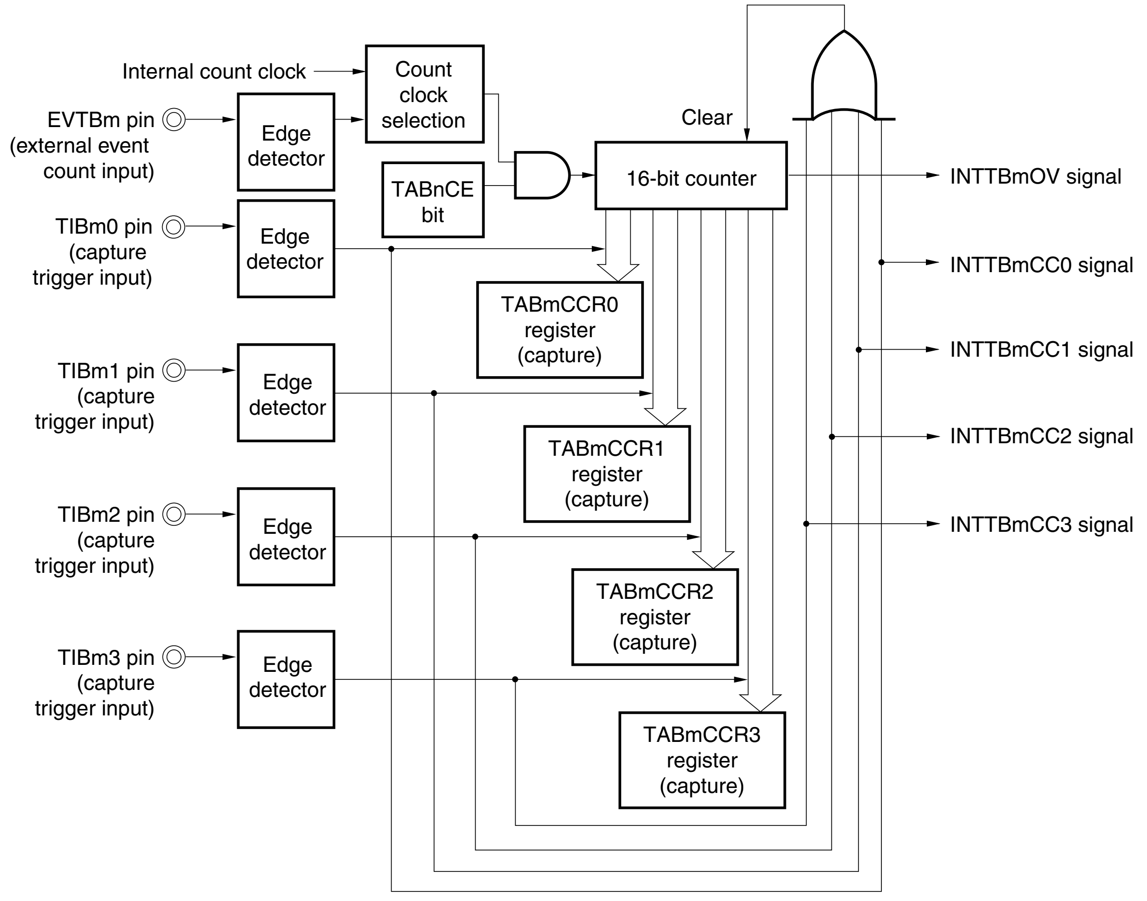
In the pulse width measurement mode, both TAB0 and TAB1 can be used in the V850E/IH4-H, but only TAB0 can be used in the V850E/IG4-H.

In the pulse width measurement mode, 16-bit timer/event counter AB starts counting when the TABmCTL0.TABmCE bit is set to 1. Each time the valid edge input to the TIBma pin has been detected, the count value of the 16-bit counter is stored in the TABmCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TABmCCRa register after a capture interrupt request signal (INTTBmCCa) occurs.

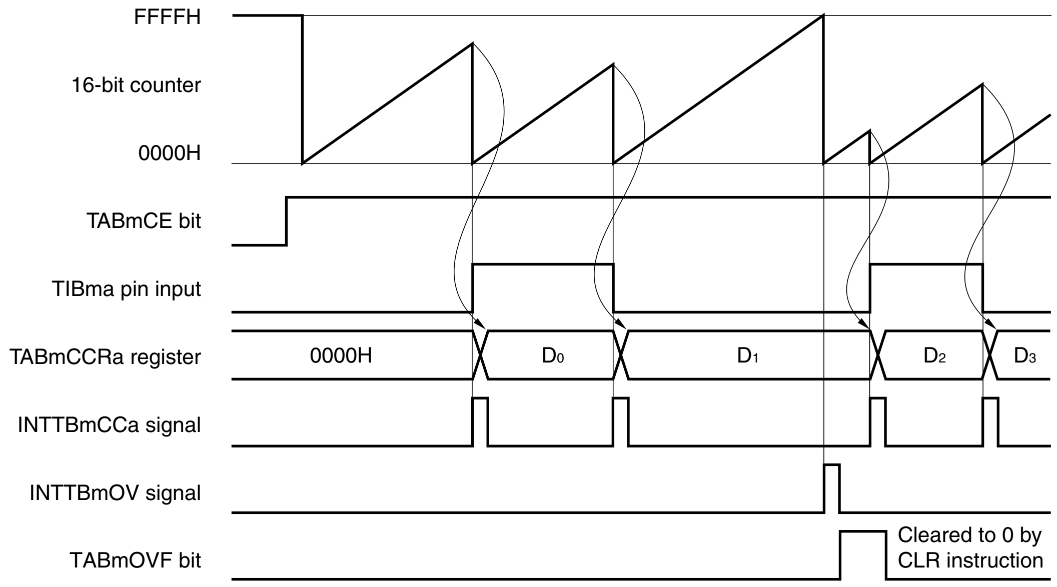
As shown in Figure 7-40, select either of the TIBm0 to TIBm3 pins as the capture trigger input pin. Specify “No edge detection” by using the TABmIOC1 register for the unused pins.

Figure 7-39. Configuration in Pulse Width Measurement Mode


Internal count clock
Count clock selection
EVTBm pin (external event count input)
Edge detector
TABnCE bit
Clear
16-bit counter
INTTBmOV signal
TIBm0 pin (capture trigger input)
Edge detector
INTTBmCC0 signal
TABmCCR0 register (capture)
INTTBmCC1 signal
TIBm1 pin (capture trigger input)
Edge detector
TABmCCR1 register (capture)
INTTBmCC2 signal
TIBm2 pin (capture trigger input)
Edge detector
INTTBmCC3 signal
TABmCCR2 register (capture)
TIBm3 pin (capture trigger input)
Edge detector
TABmCCR3 register (capture)

Remark V850E/IG4-H: m = 0, a = 0 to 3
V850E/IH4-H: m = 0, 1, a = 0 to 3

Figure 7-40. Basic Timing in Pulse Width Measurement Mode



Remark V850E/IG4-H: m = 0, a = 0 to 3
V850E/IH4-H: m = 0, 1, a = 0 to 3

When the TABmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIBma pin is later detected, the count value of the 16-bit counter is stored in the TABmCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTBmCCa) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

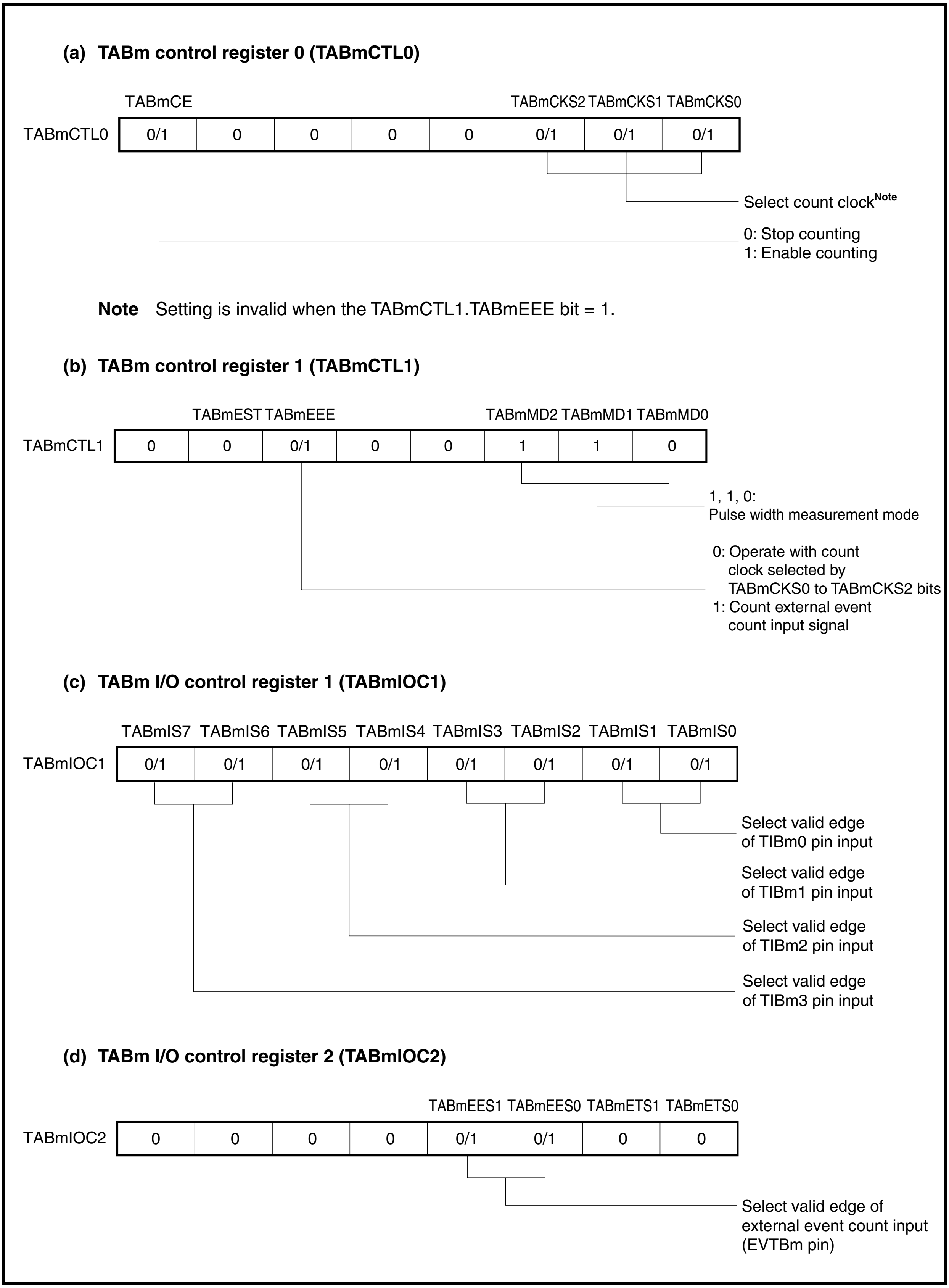
If the valid edge is not input even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTBmOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TABmOPT0.TABmOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TABmOVF bit set (1) count + Captured value) × Count clock cycle

Remark V850E/IG4-H: m = 0, a = 0 to 3
V850E/IH4-H: m = 0, 1, a = 0 to 3

Figure 7-41. Register Setting in Pulse Width Measurement Mode (1/2)



(a) TABm control register 0 (TABmCTL0)

	TABmCE					TABmCKS2	TABmCKS1	TABmCKS0
TABmCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TABmCKS2 to TABmCKS0: Select count clock[Note]

TABmCE:
0: Stop counting
1: Enable counting

Note Setting is invalid when the TABmCTL1.TABmEEE bit = 1.

(b) TABm control register 1 (TABmCTL1)

		TABmEST	TABmEEE			TABmMD2	TABmMD1	TABmMD0
TABmCTL1	0	0	0/1	0	0	1	1	0

TABmMD2 to TABmMD0:
1, 1, 0:
Pulse width measurement mode

TABmEEE:
0: Operate with count clock selected by TABmCKS0 to TABmCKS2 bits
1: Count external event count input signal

(c) TABm I/O control register 1 (TABmIOC1)

	TABmIS7	TABmIS6	TABmIS5	TABmIS4	TABmIS3	TABmIS2	TABmIS1	TABmIS0
TABmIOC1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

TABmIS1, TABmIS0: Select valid edge of TIBm0 pin input

TABmIS3, TABmIS2: Select valid edge of TIBm1 pin input

TABmIS5, TABmIS4: Select valid edge of TIBm2 pin input

TABmIS7, TABmIS6: Select valid edge of TIBm3 pin input

(d) TABm I/O control register 2 (TABmIOC2)

					TABmEES1	TABmEES0	TABmETS1	TABmETS0
TABmIOC2	0	0	0	0	0/1	0/1	0	0

TABmEES1, TABmEES0: Select valid edge of external event count input (EVTBm pin)

Figure 7-41. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TABm option register 0 (TABmOPT0)

	TABmCCS3	TABmCCS2	TABmCCS1	TABmCS0		TABmCMS	TABmCUF	TABmOVF
TABmOPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(f) TABm counter read buffer register (TABmCNT)

The value of the 16-bit counter can be read by reading the TABmCNT register.

(g) TABm capture/compare registers 0 to 3 (TABmCCR0 to TABmCCR3)

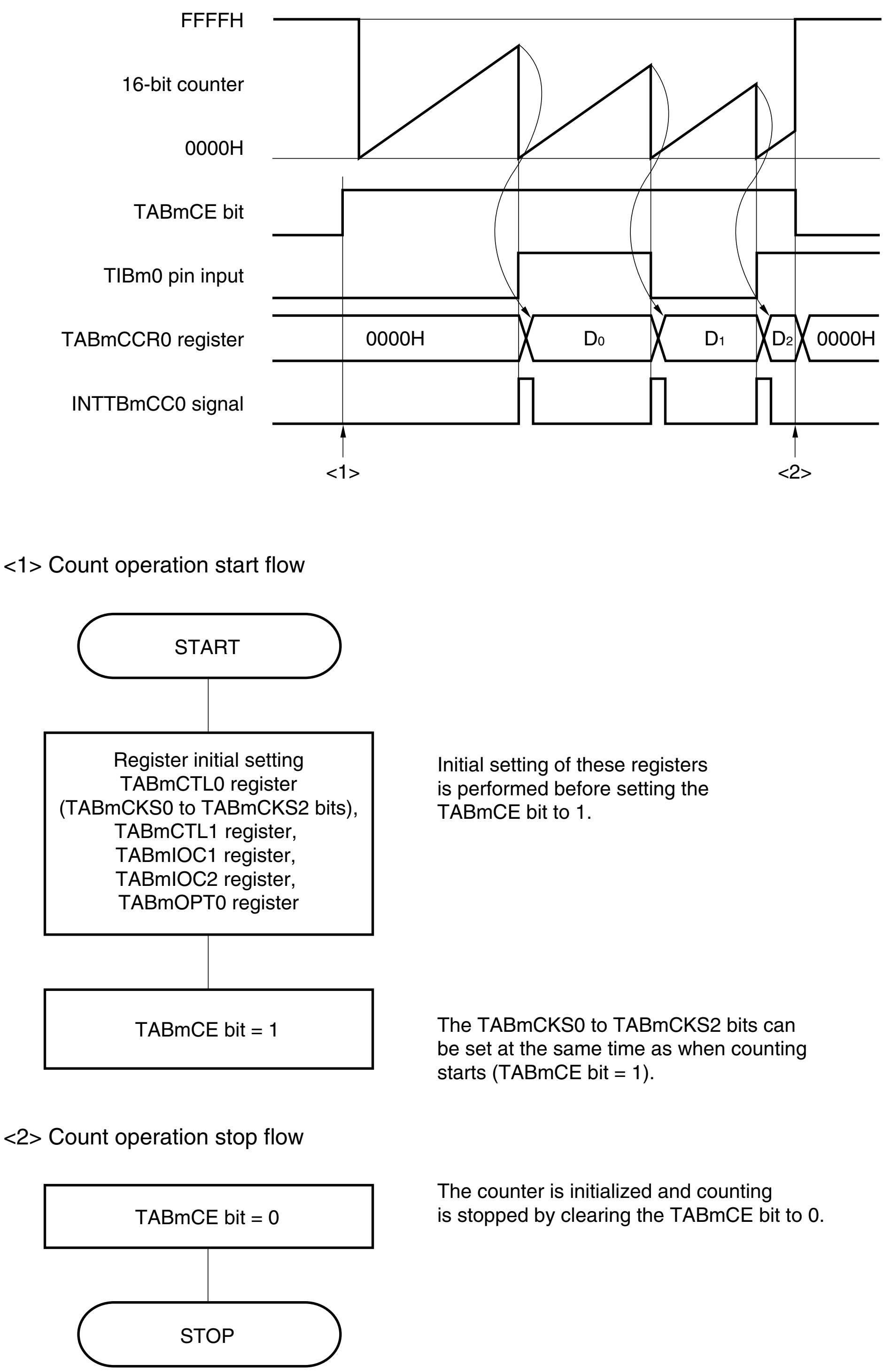
These registers store the count value of the 16-bit counter when the valid edge input to the TIBma pin is detected.

Remarks **1.** TABm I/O control register 0 (TABmIOC0) is not used in the pulse width measurement mode.

2. V850E/IG4-H: m = 0, a = 0 to 3
V850E/IH4-H: m = 0, 1, a = 0 to 3

(1) Operation flow in pulse width measurement mode

Figure 7-42. Software Processing Flow in Pulse Width Measurement Mode



<1> Count operation start flow

Initial setting of these registers is performed before setting the TABmCE bit to 1.

The TABmCKS0 to TABmCKS2 bits can be set at the same time as when counting starts (TABmCE bit = 1).

<2> Count operation stop flow

The counter is initialized and counting is stopped by clearing the TABmCE bit to 0.

Remark V850E/IG4-H: m = 0
V850E/IH4-H: m = 0, 1

(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TABmOVF bit to 0 with the CLR instruction after reading the TABmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TABmOPT0 register after reading the TABmOVF bit when it is 1.

CHAPTER 8 16-BIT TIMER/EVENT COUNTER T (TMT)

Timer T (TMT) is a 16-bit timer/event counter.

An encoder count function and other functions are added to the timer AA (TAA). However, TMT does not have a function to operate with an external event count input when it operates in the interval timer mode.

The V850E/IG4-H and V850E/IH4-H incorporate TMT0 to TMT3.

8.1 Overview

8.1.1 TMT0 and TMT1

An outline of TMT0 and TMT1 are shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: 2
- External event count input pin: 1
- External trigger input pin: 1
- Encoder input pins: 2
- Encoder clear input pin: 1
- Timer/counter: 1
- Capture/compare registers: 2
- Capture/compare match interrupt request signals: 2
- Overflow interrupt request signal: 1
- Encoder clear interrupt request signal: 1
- Timer output pins: 2

8.1.2 TMT2 and TMT3

An outline of TMT2 and TMT3 are shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: 2
- External event count input pin: 1
- External trigger input pin: 1
- Timer/counter: 1
- Capture/compare registers: 2
- Capture/compare match interrupt request signals: 2
- Overflow interrupt request signal: 1
- Timer output pins: 2

8.2 Functions

8.2.1 TMT0 and TMT1

TMT0 and TMT1 have the following functions.

- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement
- Triangular-wave PWM output mode
- Encoder count function

8.2.2 TMT2 and TMT3

TMT2 and TMT3 have the following functions.

- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement
- Triangular-wave PWM output mode

8.3 Configuration

8.3.1 TMT0 and TMT1

TMT0 and TMT1 include the following hardware.

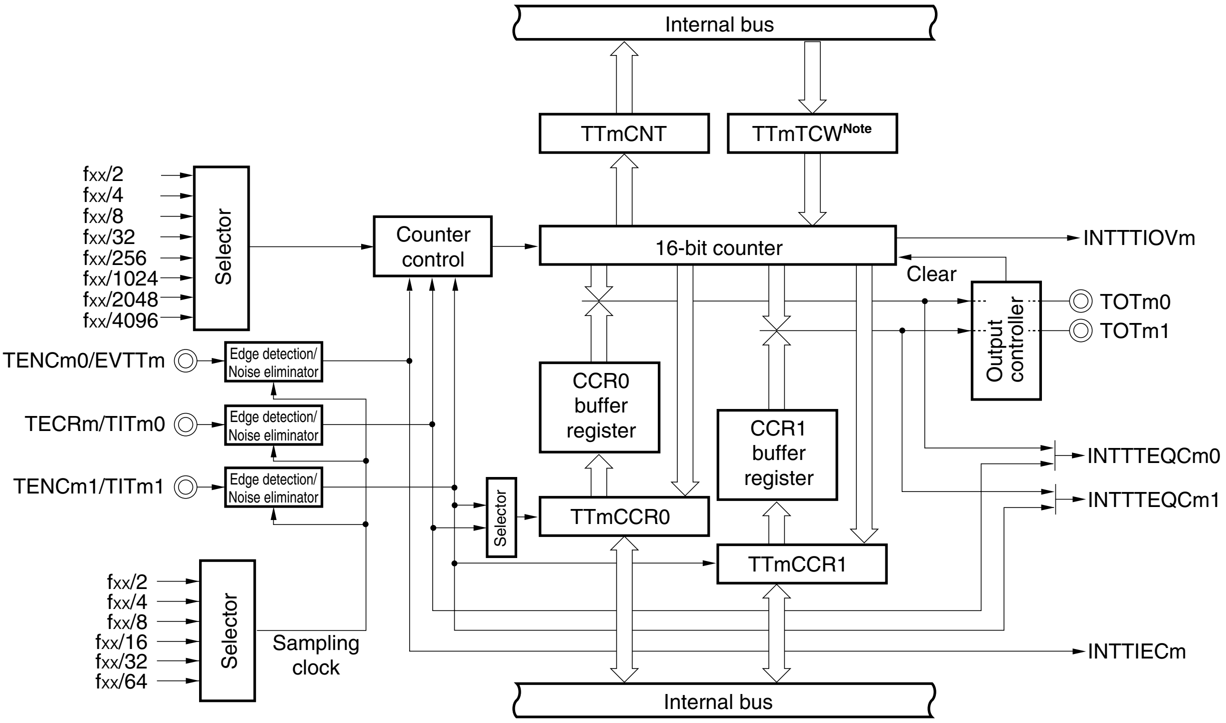
Table 8-1. Configuration of TMT0 and TMT1

Item	Configuration
Timer register	16-bit counter × 1
Registers	TMTm capture/compare registers 0, 1 (TTmCCR0, TTmCCR1) TMTm counter read buffer register (TTmCNT) TMTm counter write register (TTmTCW) CCR0 and CCR1 buffer registers
Timer input	6 in total (TITm0, TITm1, EVTTm, TENCm0, TENCm1, TECRm pins)[Note]
Timer output	2 in total (TOTm0, TOTm1 pins)[Note]
Control registers	TMTm control registers 0 to 2 (TTmCTL0 to TTmCTL2) TMTm I/O control registers 0 to 3 (TTmIOC0 to TTmIOC3) TMTm option registers 0 and 1 (TTmOPT0, TTmOPT1) TMTm capture input select register (TTISLm)

Note TITm0/TECRm pins function alternately as capture trigger input pins (TITm0), encoder clear input pins (TECRm), and timer output pins (TOTm0).
TENCm0/EVTTm pins function alternately as encoder input pins (TENCm0), external event count input pins (EVTTm), and external trigger input pins (EVTTm).
TITm1/TENCm1 pins function alternately as capture trigger input pins (TITm1), encoder input pins (TENCm1), and timer output pins (TOTm1).

Remark m = 0, 1

Figure 8-1. Block Diagram of TMT0 and TMT1



Note The initial value set from the TTmTCW register to the 16-bit counter is valid only in the encoder compare mode.
Rewrite the TTmTCW register when the TTmCTL0.TTmCE bit = 0.
The value of the TTmTCW register is transferred to the 16-bit counter when the TTmCE bit = 1.

Remarks 1. fxx: Peripheral clock

2. For the noise eliminator, see **4.6 Noise Eliminator**.

3. m = 0, 1

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TTmCNT register.

When the TTmCTL0.TTmCE bit = 0, the value of the 16-bit counter is FFFFH. If the TTmCNT register is read at this time, 0000H is read.

Reset sets the TTmCE bit to 0.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TTmCCR0 register is used as a compare register, the value written to the TTmCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCm0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is set to 0000H after reset, and the TTmCCR0 register is set to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TTmCCR1 register is used as a compare register, the value written to the TTmCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCm1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is set to 0000H after reset, and the TTmCCR1 register is set to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the TITm0, TITm1, EVTTm, TENCm0, TENCm1, and TECRm pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TTmIOC1, TTmIOC2, and TTmIOC3 registers.

(5) Output controller

This circuit controls the output of the TOTm0, and TOTm1 pins. The output controller is controlled by the TTmIOC0 registers.

(6) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

(7) Counter control

The count operation is controlled by the timer mode selected by the TTmCTL1 register.

8.3.2 TMT2 and TMT3

TMT2 and TMT3 include the following hardware.

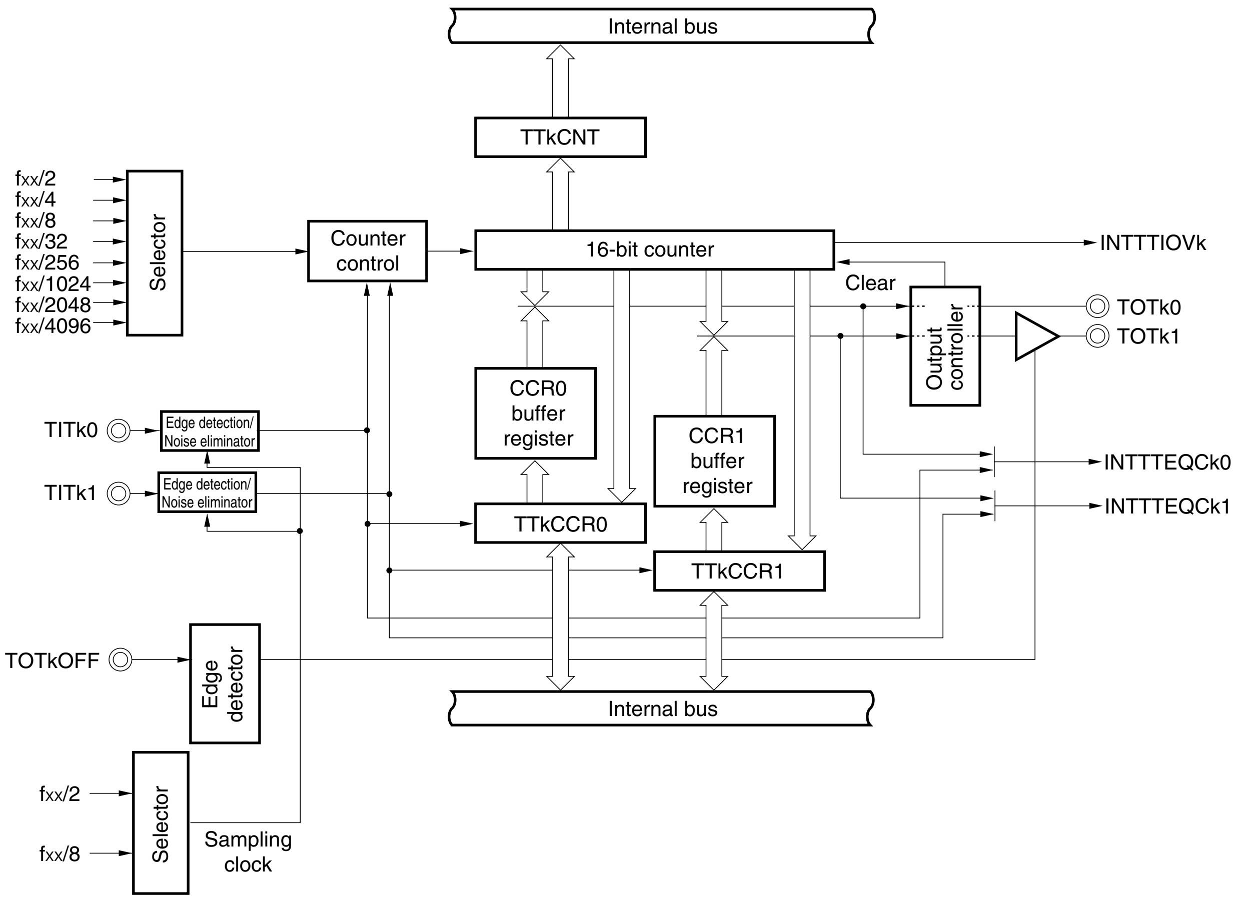
Table 8-2. Configuration of TMT2 and TMT3

Item	Configuration
Timer register	16-bit counter × 1
Registers	TMTk capture/compare registers 0, 1 (TTkCCR0, TTkCCR1) TMTk counter read buffer register (TTkCNT) CCR0 and CCR1 buffer registers
Timer input	2 in total (TITk0 and TITk1 pins)[Note]
Timer output	2 in total (TOTk0 and TOTk1 pins)[Note]
Control registers	TMTk control registers 0, 1 (TTkCTL0, TTkCTL1) TMTk I/O control registers 0 to 2 (TTkIOC0 to TTkIOC2) TMTk option register 0 (TTkOPT0)

Note The TITk0 pin is also used for the capture trigger input signal, the external event count input signal, the external trigger input signal, and as the timer output pin (TOTk0).
The TITk1 pin is also used for the capture trigger input signal and as the timer output pin (TOTk1).

Remark k = 2, 3

Figure 8-2. Block Diagram of TMT2 and TMT3



Remarks 1. fxx: Peripheral clock

2. For the TOTkOFF pin, see **10.3 (6) High-impedance output control registers 00, 01, 10, 11, 20, 21, 30, 31, 40, 41, 50, 51, 60, 61, 70, 71, 80, 81, 90, 91, 100, 101, 110, 111, 120, 121 (HZAyCTL0, HZAyCTL1)**.

3. For the noise eliminator, see **4.6 Noise Eliminator**.

4. k = 2, 3

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TTkCNT register.

When the TTkCTL0.TTkCE bit = 0, the value of the 16-bit counter is FFFFH. If the TTkCNT register is read at this time, 0000H is read.

Reset sets the TTkCE bit to 0.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TTkCCR0 register is used as a compare register, the value written to the TTkCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCk0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is set to 0000H after reset, and the TTkCCR0 register is set to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TTkCCR1 register is used as a compare register, the value written to the TTkCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCk1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is set to 0000H after reset, and the TTkCCR1 register is set to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the TITk0 and TITk1 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TTkIOC1, TTkIOC2 registers.

(5) Output controller

This circuit controls the output of the TOTk0 and TOTk1 pins. The output controller is controlled by the TTkIOC0 registers.

(6) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

(7) Counter control

The count operation is controlled by the timer mode selected by the TTkCTL1 register.

8.4 Registers

(1) TMTn control register 0 (TTnCTL0)

The TTnCTL0 register is an 8-bit register that controls the operation of TMTn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

The same value can always be written to the TTnCTL0 register by software.

After reset: 00H R/W Address: TT0CTL0 FFFFF580H, TT1CTL0 FFFFF5C0H,
TT2CTL0 FFFFF780H, TT3CTL0 FFFFF7C0H

	<7>	6	5	4	3	2	1	0
TTnCTL0 (n = 0 to 3)	TTnCE	0	0	0	0	TTnCKS2	TTnCKS1	TTnCKS0

TTnCE	TMTn operation control
0	TMTn operation disabled (TMTn reset asynchronously[Note])
1	TMTn operation enabled. TMTn operation start

TTnCKS2	TTnCKS1	TTnCKS0	Internal count clock selection
0	0	0	fxx/2
0	0	1	fxx/4
0	1	0	fxx/8
0	1	1	fxx/32
1	0	0	fxx/256
1	0	1	fxx/1024
1	1	0	fxx/2048
1	1	1	fxx/4096

Note The TTnOPT0.TTnOVF bit and the 16-bit counter are reset simultaneously. Moreover, timer outputs (TOTn0 and TOTn1 pins) are reset to the TTnIOC0 register set status at the same time as the 16-bit counter is reset.

Cautions 1. Set the TTnCKS2 to TTnCKS0 bits when the TTnCE bit = 0.
When the value of the TTnCE bit is changed from 0 to 1, the TTnCKS2 to TTnCKS0 bits can be set simultaneously.
2. Be sure to set bits 3 to 6 to “0”.

Remark fxx: Peripheral clock

(2) TMTn control register 1 (TTnCTL1)

The TTnCTL1 register is an 8-bit register that controls the TMTn operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TT0CTL1 FFFFF581H, TT1CTL1 FFFFF5C1H,
TT2CTL1 FFFFF781H, TT3CTL1 FFFFF7C1H

	7	6	5	4	3	2	1	0
TTnCTL1 (n = 0 to 3)	0	TTnEST	TTnEEE	0	TTnMD3	TTnMD2	TTnMD1	TTnMD0

TTnEST	Software trigger control
1	–
	Generate a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TTnEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TTnEST bit as the trigger.
The read value of the TTnEST bit is always 0.	

TTnEEE	Count clock selection
0	Disable operation with external event count input[Note 1]. (Perform counting with the count clock selected by the TTnCTL0.TTnCKS0 to TTnCTL0.TTnCKS2 bits.)
1	Enable operation with external event count input[Note 1]. (Perform counting at the valid edge of the external event count input signal[Note 1].)
The TTnEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TTnMD3	TTnMD2	TTnMD1	TTnMD0	Timer mode selection
0	0	0	0	Interval timer mode
0	0	0	1	External event count mode
0	0	1	0	External trigger pulse output mode
0	0	1	1	One-shot pulse output mode
0	1	0	0	PWM output mode
0	1	0	1	Free-running timer mode
0	1	1	0	Pulse width measurement mode
0	1	1	1	Triangular-wave PWM output mode
1	0	0	0	Encoder compare mode[Note 2]
Other than above				Setting prohibited

Notes 1. TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

2. Setting to the encoder compare mode for TMT2 and TMT3 is prohibited.

(2/2)

Cautions 1. The TTnEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. The TTnEEE bit is valid only in the interval timer mode, external trigger pulse output mode, one-shot pulse output mode, PWM output mode, free-running timer mode, pulse width measurement mode, or triangular-wave PWM output mode. In any other mode, writing 1 to this bit is ignored.

3. The external event count input (the EVTTm pin in the case of TMT0 and TMT1, and the TITk0 pin in the case of TMT2 and TMT3) is selected in the external event count mode and the encoder inputs (TENCm0 and TENCm1) are selected in the encoder compare mode (TMT2 and TMT3 only), regardless of the value of the TTmEEE bit (m = 0 or 1, k = 2 or 3).

4. Set the TTmEEE and TTnMD3 to TTnMD0 bits when the TTnCTL0.TTnCE bit = 0. (The same value can be written when the TTnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TTnCE bit = 1. If rewriting was mistakenly performed, clear the TTnCE bit to 0 and then set the bits again.

5. Be sure to set bits 4 and 7 to “0”.

(3) TMTm control register 2 (TTmCTL2)

The TTmCTL2 register is an 8-bit register that controls the encoder count function operation.

The TTmCTL2 register is valid only in the encoder compare mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution For details of each bit of the TTmCTL2 register, see 8.6.9 (5) Controlling bits of TTmCTL2 register.

(1/2)

After reset: 00H R/W Address: TT0CTL2 FFFFF582H, TT1CTL2 FFFFF5C2H

	7	6	5	4	3	2	1	0
TTmCTL2 (m = 0, 1)	TTmECC	0	0	TTmLDE	TTmECM1	TTmECM0	TTmUDS1	TTmUDS0

TTmECC	Encoder counter control
0	Normal operation
1	Holds count value of 16-bit counter when TTmCTL0.TTmCE bit = 0.

TTmLDE	Transfer setting to 16-bit counter
0	Disables transfer of set value of TTmCCR0 to 16-bit counter in case of underflow.
1	Enables transfer of set value of TTmCCR0 to 16-bit counter in case of underflow.

TTmECM1	Control of encoder clear operation 1
0	The 16-bit counter is not cleared to 0000H when its count value matches value of CCR1 register.
1	The 16-bit counter is cleared to 0000H when its count value matches value of CCR1 register.

TTmECM0	Control of encoder clear operation 0
0	The 16-bit counter is not cleared to 0000H when its count value matches value of CCR0 register.
1	The 16-bit counter is cleared to 0000H when its count value matches value of CCR0 register.

(2/2)

TTmUDS1	TTmUDS0	Up/down count selection
0	0	When valid edge of TENCm0 input is detected Counts down when TENCm1 = high level. Counts up when TENCm1 = low level.
0	1	Counts up when valid edge of TENCm0 input is detected. Counts down when valid edge of TENCm1 input is detected.
1	0	Counts down when rising edge of TENCm0 input is detected. Counts up when falling edge of TENCm0 input is detected. However, count operation is performed only when TENCm1 = low level.
1	1	Both rising and falling edges of TENCm0 and TENCm1 are detected. Count operation is automatically identified by combination of edge detection and level detection.

Cautions 1. The TTmECC bit is valid only in the encoder compare mode. In any other mode, writing “1” to this bit is ignored.

If the TTmCTL0.TTmCE bit is cleared to 0 while the TTmECC bit = 1, the values of the timer/counter and capture registers (TTmCCR0 and TTmCCR1), and the TTmOPT1, TTmEUF, TTmEOF, and TTmESF flags are retained.

If the TTmCE bit is set from 0 to 1 when the TTmECC bit = 1, the value of the TTmTCW register is not transferred to the 16-bit counter.

2. The TTmLDE bit is valid only when the TTmECM1 and TTmECM0 bits = 00, 01. Writing “1” to this bit is ignored when the TTmECM1 and TTmECM0 bits = 10, 11.

3. The edge detection of the TENCm0 and TENCm1 inputs specified by the TTmIOC3.TTmEIS1 and TTmIOC3.TTmEIS0 bits is invalid and fixed to both the rising and falling edges when the TTmUDS1 and TTmUDS0 bits = 10, 11.

4. Set the TTmLDE, TTmECM1, TTmECM0, TTmUDS1, and TTmUDS0 bits when the TTmCTL0.TTmCE bit = 0 (the same value can be written to these bits when the TTmCE bit = 1). If the value of these bits is changed when the TTmCE bit = 1, the operation cannot be guaranteed. If it is changed by mistake, clear the TTmCE bit and then set the correct value.

5. Be sure to set bits 5 and 6 to “0”.

(4) TMTn I/O control register 0 (TTnIOC0)

The TTnIOC0 register is an 8-bit register that controls the timer output (TOTn0, TOTn1 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TT0IOC0 FFFFF583H, TT1IOC0 FFFFF5C3H,
TT2IOC0 FFFFF783H, TT3IOC0 FFFFF7C3H

	7	6	5	4	3	<2>	1	<0>
TTnIOC0 (n = 0 to 3)	0	0	0	0	TTnOL1	TTnOE1	TTnOL0	TTnOE0

TTnOL1	TOTn1 pin output level setting[Note]
0	TOTn1 pin starts output at high level.
1	TOTn1 pin starts output at low level.

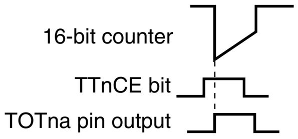
TTnOE1	TOTn1 pin output setting
0	Timer output prohibited • Low level is output from the TOTn1 pin when the TTnOL1 bit = 0. • High level is output from the TOTn1 pin when the TTnOL1 bit = 1.
1	Timer output enabled (A pulse is output from the TOTn1 pin.)

TTnOL0	TOTn0 pin output level setting[Note]
0	TOTn0 pin starts output at high level.
1	TOTn0 pin starts output at low level.

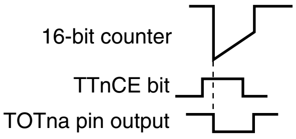
TTnOE0	TOTn0 pin output setting
0	Timer output prohibited • Low level is output from the TOTn0 pin when the TTnOL0 bit = 0. • High level is output from the TOTn0 pin when the TTnOL0 bit = 1.
1	Timer output enabled (A pulse is output from the TOTn0 pin.)

Note The output level of the timer output pins (TOTn0, TOTn1) in modes other than the triangular-wave PWM output mode, which is specified by the TTnOLa bit, is as follows (a = 0, 1).

• When TTnOLa bit = 0



• When TTnOLa bit = 1



For the output level in the triangular-wave PWM output mode, see **Figure 8-51 Basic Timing in Triangular-Wave PWM Output Mode**.

Cautions 1. If the setting of the TTnIOC0 register is changed when TOTn0 and TOTn1 outputs are set for the port mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.

2. Rewrite the TTnOL1, TTnOE1, TTnOL0, and TTnOE0 bits when the TTnCTL0.TTnCE bit = 0. (The same value can be written when the TTnCE bit = 1.) If rewriting was mistakenly performed, clear the TTnCE bit to 0 and then set the bits again.

3. Even if the TTnOL0 or TTnOL1 bit is manipulated when the TTnCE, TTnOE0, and TTnOE1 bits are 0, the output level of the TOTn0 and TOTn1 pins changes.

(5) TMTn I/O control register 1 (TTnIOC1)

The TTnIOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TITn0, TITn1 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TT0IOC1 FFFFF584H, TT1IOC1 FFFFF5C4H,
TT2IOC1 FFFFF784H, TT3IOC1 FFFFF7C4H

	7	6	5	4	3	2	1	0
TTnIOC1 (n = 0 to 3)	0	0	0	0	TTnIS3	TTnIS2	TTnIS1	TTnIS0

TTnIS3	TTnIS2	Capture trigger input signal (TITn1 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TTnIS1	TTnIS0	Capture trigger input signal (TITn0 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TTnIS3 to TTnIS0 bits when the TTnCTL0.TTnCE bit = 0.
(The same value can be written when the TTnCE bit = 1.) If rewriting was mistakenly performed, clear the TTnCE bit to 0 and then set the bits again.

2. The TTnIS3 and TTnIS2 bits are valid only in the free-running timer mode (only when the TTnOPT0.TTnCCS1 bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not possible.
The TTnIS1 and TTnIS0 bits are valid only in the free-running timer mode (only when the TTnOPT0. TTnCCS0 bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not possible.

(6) TMTn I/O control register 2 (TTnIOC2)

The TTnIOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (the EVTTm pin in the case of TMT0 and TMT1, and the TITk0 pin in the case of TMT2 and TMT3) and external trigger input signal (the EVTTm pin in the case of TMT0 and TMT1, and the TITk0 pin in the case of TMT2 and TMT3) (m = 0, 1, k = 2, 3).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TT0IOC2 FFFFF585H, TT1IOC2 FFFFF5C5H
TT2IOC2 FFFFF785H, TT3IOC2 FFFFF7C5H

	7	6	5	4	3	2	1	0
TTnIOC2 (n = 0 to 3)	0	0	0	0	TTnEES1	TTnEES0	TTnETS1	TTnETS0

TTnEES1	TTnEES0	External event count input signal[Note] valid edge setting
0	0	No edge detection (external event count invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TTnETS1	TTnETS0	External trigger input signal[Note] valid edge setting
0	0	No edge detection (external trigger invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Note TMT0 and TMT1: EVTTm pin
TMT2 and TMT3: TITk0 pin

Cautions 1. Rewrite the TTnEES1, TTnEES0, TTnETS1, and TTnETS0 bits when the TTnCTL0.TTnCE bit = 0. (The same value can be written when the TTnCE bit = 1.) If rewriting was mistakenly performed, clear the TTnCE bit to 0 and then set the bits again.

2. The TTnEES1 and TTnEES0 bits are valid only when the TTnCTL1.TTnEEE bit = 1 or when the external event count mode (the TTnCTL1.TTnMD3 to TTnCTL1.TTnMD0 bits = 0001) has been set.

3. The TTnETS1 and TTnETS0 bits are valid only in the external trigger pulse output mode or one-shot pulse output mode.

(7) TMTm I/O control register 3 (TTmIOC3)

The TTmIOC3 register is an 8-bit register that controls the encoder clear function operation.

The TTmIOC3 register is valid only in the encoder compare mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TT0IOC3 FFFFF586H, TT1IOC3 FFFFF5C6H

	7	6	5	4	3	2	1	0
TTmIOC3 (m = 0, 1)	TTmSCE	TTmZCL	TTmBCL	TTmACL	TTmECS1	TTmECS0	TTmEIS1	TTmEIS0

TTmSCE	Encoder clear selection
0	Clears 16-bit counter on detection of edge of encoder clear signal (TECRm pin).
1	Clears 16-bit counter on detection of clear level condition of the TENCm0, TENCm1, and TECRm pins.

- Clears 16-bit counter to 0000H when valid edge of TECRm pin specified by the TTmECS1 and TTmECS0 bits is detected when the TTmSCE bit = 0.
- Clears 16-bit counter to 0000H when clear level conditions of the TTmZCL, TTmBCL, and TTmACL bits match input levels of the TECRm, TENCm1, and TENCm0 pins when TTmSCE bit = 1.
- Setting of the TTmZCL, TTmBCL, and TTmACL bits is valid and that of the TTmECS1 and TTmECS0 bits is invalid when the TTmSCE bit = 1.
  Encoder clear interrupt request signal (INTTIECm) is not generated.
- Setting of the TTmZCL, TTmBCL, and TTmACL bits is invalid and setting of the TTmECS1 and TTmECS0 bits is valid when the TTmSCE bit = 0.
  The INTTIECm signal is generated when valid edge specified by the TTmECS1 and TTmECS0 bits is detected.
- Be sure to set the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits to 10 or 11 when the TTmSCE bit = 1.
  Operation is not guaranteed if the TTmUDS1 and TTmUDS0 bits = 00 or 01 and the TTmSCE bit = 1.

TTmZCL	Clear level selection of encoder clear signal (TECRm pin)
0	Clears low level of the TECRm pin.
1	Clears high level of the TECRm pin.

Setting of the TTmZCL bit is valid only when the TTmSCE bit = 1.

TTmBCL	Clear level selection of encoder input signal (TENCm1 pin)
0	Clears low level of the TENCm1 pin.
1	Clears high level of the TENCm1 pin.

Setting of the TTmBCL bit is valid only when the TTmSCE bit = 1.

TTmACL	Clear level selection of encoder input signal (TENCm0 pin)
0	Clears low level of the TENCm0 pin.
1	Clears high level of the TENCm0 pin.

Setting of the TTmACL bit is valid only when the TTmSCE bit = 1.

(2/2)

TTmECS1	TTmECS0	Valid edge setting of encoder clear signal (TECRm pin)
0	0	Detects no edge (clearing encoder is invalid).
0	1	Detects rising edge.
1	0	Detects falling edge.
1	1	Detects both edges.

TTmEIS1	TTmEIS0	Valid edge setting of encoder input signals (TENCm0, TENCm1 pins)
0	0	Detects no edge (inputting encoder is invalid).
0	1	Detects rising edge.
1	0	Detects falling edge.
1	1	Detects both edges.

Cautions 1. Rewrite the TTmSCE, TTmZCL, TTmBCL, TTmACL, TTmECS1, TTmECS0, TTmEIS1, and TTmEIS0 bits when the TTmCTL0.TTmCE bit = 0. (The same value can be written to these bits when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set these bits again.

2. The TTmECS1 and TTmECS0 bits are valid only when the TTmSCE bit = 0 and the encoder compare mode is set.

3. The TTmEIS1 and TTmEIS0 bits are valid only when the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits = 00 or 01.

(8) TMTn option register 0 (TTnOPT0)

The TTnOPT0 register is an 8-bit register that sets the capture/compare operation and detects overflow.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TT0OPT0 FFFFF587H, TT1OPT0 FFFFF5C7H
TT2OPT0 FFFFF787H, TT3OPT0 FFFFF7C7H

	7	6	5	4	3	2	1	<0>
TTnOPT0 (n = 0 to 3)	0	0	TTnCCS1	TTnCCS0	0	0	0	TTnOVF

TTnCCS1	TTnCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by the TTnCTL0.TTnCE bit = 0)
The TTnCCS1 bit setting is valid only in the free-running timer mode.	

TTnCCS0	TTnCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by the TTnCTL0.TTnCE bit = 0)
The TTnCCS0 bit setting is valid only in the free-running timer mode.	

TTnOVF	TMTn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	0 written to TTnOVF bit or TTnCTL0.TTnCE bit = 0
• The TTnOVF bit is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode. • An overflow interrupt request signal (INTTTIOVn) is generated at the same time that the TTnOVF bit is set to 1. The INTTTIOVn signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. • The TTnOVF bit is not cleared to 0 even when the TTnOVF bit or the TTnOPT0 register are read when the TTnOVF bit = 1. • Before clearing the TTnOVF bit to 0 after generation of the INTTTIOVn signal, be sure to confirm (by reading) that the TTnOVF bit is set to 1. • The TTnOVF bit can be both read and written, but the TTnOVF bit cannot be set to 1 by software. Writing 1 has no effect on the operation of TMTn.	

Cautions 1. Rewrite the TTnCCS1 and TTnCCS0 bits when the TTnCE bit = 0. (The same value can be written when the TTnCE bit = 1.) If rewriting was mistakenly performed, clear the TTnCE bit to 0 and then set these bits again.

2. Be sure to set bits 1 to 3, 6, and 7 to “0”.

(9) TMTm option register 1 (TTmOPT1)

The TTmOPT1 register is an 8-bit register that detects the overflow, underflow, and count-up/down operation of the encoder count function.

The TTmOPT1 register is valid only in the encoder compare mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

This register can be rewritten even when the TTmCTL0.TTmCE bit = 1.

(1/2)

After reset: 00H R/W Address: TT0OPT1 FFFFF588H, TT1OPT1 FFFFF5C8H

	7	6	5	4	3	<2>	<1>	<0>
TTmOPT1 (m = 0, 1)	0	0	0	0	0	TTmEUF	TTmEOF	TTmESF

TTmEUF	TMTm underflow detection flag
Set (1)	Underflow occurs.
Reset (0)	Cleared by writing to TTmEUF bit or when TTmCTL0.TTmCE bit = 0

- The TTmEUF bit is set to 1 when 16-bit counter underflows from 0000H to FFFFH in encoder compare mode.
- When the TTmCTL2.TTmLDE bit = 1, TTmEUF bit is set to 1 when value of 16-bit counter is changed from 0000H to set value of the TTmCCR0 register.
- Overflow interrupt request signal (INTTTIOVm) is generated as soon as the TTmEUF bit is set to 1.
- The TTmEUF bit is not cleared to 0 even if the TTmEUF bit or TTmOPT1 register is read when the TTmEUF bit = 1.
- Status of the TTmEUF bit is retained even if the TTmCTL0.TTmCE bit is cleared to 0 when the TTmCTL2.TTmECC bit = 1.
- Before clearing the TTmEUF bit to 0 after the INTTTIOVm signal is generated, be sure to confirm (read) that the TTmEUF bit is set to 1.
- The TTmEUF bit can be read or written, but it cannot be set to 1 by software. Setting this bit to 1 does not affect operation of TMTm.

(2/2)

TTmEOF	Overflow detection flag for TMTm encoder function
Set (1)	Overflow occurs.
Reset (0)	Cleared by writing 0 to the TTmEOF bit or when the TTmCTL0.TTmCE bit = 0

- The TTmEOF bit is set to 1 when 16-bit counter overflows from FFFFH to 0000H in encoder compare mode.
- As soon as the TTmEOF bit has been set to 1, an overflow interrupt request signal (INTTTIOVm) is generated. At this time, the TTmOPT0.TTmOVF bit is not set to 1.
- The TTmEOF bit is not cleared to 0 even if the TTmEOF bit or TTmOPT1 register is read when the TTmEOF bit = 1.
- Status of the TTmEOF bit is retained even if the TTmCTL0.TTmCE bit is cleared to 0 when the TTmCTL2.TTmECC bit = 1.
- Before clearing the TTmEOF bit to 0 after the INTTTIOVm signal is generated, be sure to confirm (read) that the TTmEOF bit is set to 1.
- The TTmEOF bit can be read or written, but it cannot be set to 1 by software. Writing 1 to this bit does not affect operation of TMTm.

TTmESF	TMTm count-up/-down operation status detection flag
0	TMTm is counting up.
1	TMTm is counting down.

- This bit is cleared to 0 if the TTmCTL0.TTmCE bit = 0 when the TTmCTL2.TTmECC bit = 0.
- Status of the TTmESF bit is retained even if the TTmCE bit = 0 when the TTmECC bit = 1.

Caution Be sure to set bits 3 to 7 to “0”.

(10) TMTm capture input select register (TTISLm)

The TTISLm register is used to select which of TITm0 or TITm1 pin is used to input a capture trigger input signal when the TTmCCR0 register is used as a capture register.

This register can be read or written in 8-bit or 1-bit units.

Reset makes this register undefined.

After reset: Undefined R/W Address: TTISL0 FFFFF5A4H, TTISL1 FFFFF5A6H

	7	6	5	4	3	2	1	0
TTISLm (m = 0, 1)	0	0	0	0	0	0	0	TTISLm

TTISLm	Capture trigger input signal selection of TTmCCR0 register
0	TITm0 input
1	TITm1 input

(11) TMTn capture/compare register 0 (TTnCCR0)

The TTnCCR0 register is a 16-bit register that can be used as a capture register or compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TTnOPT0.TTnCCS0 bit. In the pulse width measurement mode, the TTnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TTnCCR0 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Remark n = 0 to 3

After reset: 0000H R/W Address: TT0CCR0 FFFFF58AH, TT1CCR0 FFFFF5CAH,
TT2CCR0 FFFFF78AH, TT3CCR0 FFFFF7CAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR0 (n = 0 to 3)																

(a) Function as compare register

The TTnCCR0 register can be rewritten even when the TTnCTL0.TTnCE bit = 1.

The set value of the TTnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCn0) is generated. If TOTn0 pin output is enabled at this time, the output of the TOTn0 pin is inverted.

When the TTnCCR0 register is used as a cycle register in the interval timer mode, external event count mode, external trigger pulse output mode, one-shot pulse output mode, PWM output mode, and triangular-wave PWM output mode or the TTmCCR0 register is used as a cycle register in the encoder compare mode, the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

The compare register is not cleared by setting the TTnCTL0.TTnCE bit to 0.

(b) Function as capture register

When the TTnCCR0 register is used as a capture register in the free-running timer mode (when the TTnCCR0 register is used as a capture register), the count value of the 16-bit counter is stored in the TTnCCR0 register if the valid edge of the capture trigger input pin (TITn0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TTnCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TITn0 pin) is detected.

Even if the capture operation and reading the TTnCCR0 register conflict, the correct value of the TTnCCR0 register can be read.

The capture register is cleared by setting the TTnCTL0.TTnCE bit to 0.

Remark n = 0 to 3
m = 0, 1

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 8-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note 2]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note 2]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None
Triangular-wave PWM output	Compare register	Batch write[Note 2]
Encoder compare[Note 1]	Compare register	Anytime write

Notes 1. TMT0, TMT1 only.
2. Writing to the TTnCCR1 register is the trigger.

Remark For anytime write and batch write, see **8.6 (3) Anytime write and batch write**.

(12) TMTn capture/compare register 1 (TTnCCR1)

The TTnCCR1 register is a 16-bit register that can be used as a capture register or compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TTnOPT0.TTnCCS1 bit. In the pulse width measurement mode, the TTnCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TTnCCR1 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Remark n = 0 to 3

After reset: 0000H R/W Address: TT0CCR1 FFFFF58CH, TT1CCR1 FFFFF5CCH
TT2CCR1 FFFFF78CH, TT3CCR1 FFFFF7CCH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR1 (n = 0 to 3)																

(a) Function as compare register

The TTnCCR1 register can be rewritten even when the TTnCTL0.TTnCE bit = 1.

The set value of the TTnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCn1) is generated. If TOTn1 pin output is enabled at this time, the output of the TOTn1 pin is inverted.

The compare register is not cleared by setting the TTnCTL0.TTnCE bit to 0.

(b) Function as capture register

When the TTnCCR1 register is used as a capture register in the free-running timer mode (when the TTnCCR1 register is used as a capture register), the count value of the 16-bit counter is stored in the TTnCCR1 register if the valid edge of the capture trigger input pin (TITn1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TTnCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TITn1 pin) is detected.

Even if the capture operation and reading the TTnCCR1 register conflict, the correct value of the TTnCCR1 register can be read.

The capture register is cleared by setting the TTnCTL0.TTnCE bit to 0.

Remark n = 0 to 3

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 8-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note 2]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note 2]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None
Triangular-wave PWM output	Compare register	Batch write[Note 2]
Encoder compare[Note 1]	Compare register	Anytime write

Notes 1. TMT0 and TMT1 only.

2. Writing to the TTnCCR1 register is the trigger.

Remark For anytime write and batch write, see **8.6 (3) Anytime write and batch write**.

(13) TMTm counter write register (TTmTCW)

The TTmTCW register is used to set the initial value of the 16-bit counter.

The TTmTCW register is valid only in the encoder compare mode.

This register can be read or written in 16-bit units.

Rewrite the TTmTCW register when the TTmCTL0.TTmCE bit = 0.

The value of the TTmTCW register is transferred to the 16-bit counter when the TTmCE bit is set (1).

Reset sets this register to 0000H.

After reset: 0000H R/W Address: TT0TCW FFFFF590H, TT1TCW FFFFF5D0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTmTCW (m = 0, 1)																

(14) TMTn counter read buffer register (TTnCNT)

The TTnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TTnCTL0.TTnCE bit = 1, the count value of the 16-bit timer can be read.

This register is read-only, in 16-bit units.

The value of the TTmCNT register is set to 0000H when the TTmCTL2.TTmECC and TTmCE bits = 0. If the TTmCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read. The TTmCNT register is not set to 0000H but the previous value is read when the TTmECC bit = 1 and TTmCE bit = 0.

The TTmECC and TTmCE bits are set to 0 after reset, and the value of the TTmCNT register is set to 0000H.

After reset: 0000H R Address: TT0CNT FFFFF58EH, TT1CNT FFFFF5CEH,
TT2CNT FFFFF78EH, TT3CNT FFFFF7CEH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCNT (n = 0 to 3)																

8.5 Timer Output Operations

The following table shows the operations and output levels of the TOTn0 and TOTn1 pins.

Table 8-5. Timer Output Control in Each Mode

Operation Mode	TOTn1 Pin	TOTn0 Pin
Interval timer mode	PWM output	
External event count mode	None	
External trigger pulse output mode	External trigger pulse output	PWM output
One-shot pulse output mode	One-shot pulse output	
PWM output mode	PWM output	
Free-running timer mode	PWM output (only when compare function is used)	
Pulse width measurement mode	None	
Triangular-wave PWM output mode	Triangular-wave PWM output	
Encoder compare mode[Note]	None	

Note TMT0 and TMT1 only.

Remark n = 0 to 3

Table 8-6. Truth Table of TOTn0 and TOTn1 Pins Under Control of Timer Output Control Bits

TTnIOC0.TTnOLa Bit	TTnIOC0.TTnOEa Bit	TTnCTL0.TTnCE Bit	Level of TOTna Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark n = 0, 1
a = 0, 1

8.6 Operation

The functions of TMTn that can be implemented differ from one channel to another. The functions of each channel are shown below (n = 0 to 3).

Table 8-7. Specifications of TMT0 and TMT1 in Each Mode

Operation	TTmCTL1.TTmEST Bit (Software Trigger Bit)	EVTTm Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode	Valid	Valid	Compare only	Batch write
One-shot pulse output mode	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switchable	Anytime write
Pulse width measurement mode	Invalid	Invalid	Capture only	Not applicable
Triangular-wave PWM output mode	Invalid	Invalid	Compare only	Batch write
Encoder compare mode	Invalid	Invalid	Compare only	Anytime write

Remark m = 0, 1

Table 8-8. Specifications of TMT2 and TMT3 in Each Mode

Operation	TTkCTL1.TTkEST Bit (Software Trigger Bit)	TITk0 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode	Valid	Valid	Compare only	Batch write
One-shot pulse output mode	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switchable	Anytime write
Pulse width measurement mode	Invalid	Invalid	Capture only	Not applicable
Triangular-wave PWM output mode	Invalid	Invalid	Compare only	Batch write

Remark k = 2, 3

(1) Basic counter operation of TMT0 and TMT1

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

Remark m = 0, 1

(a) Counter start operation

- In external event count mode
  When the TTmCTL0.TTmCE bit is set from 0 to 1, the 16-bit counter is set to 0000H.
  After that, it counts up to 0001H, 0002H, 0003H, … each time the valid edge of external event count input (EVTTm) is detected.
- Encoder compare mode
  The count operation is controlled by TENCm0 and TENCm1 phases.
  When the 16-bit counter initial setting is performed by transferring the set value of the TTmTCW register to the 16-bit counter and the count operation is started. (When the TTmCTL2.TTmECC bit = 0, the TTmTCW register set value is transferred to the 16-bit counter at the timing when the TTmCTL0.TTmCE bit changes from 0 to 1.)
- Triangular-wave PWM mode
  The 16-bit counter starts counting from the initial value FFFFH.
  It counts up FFFFH, 0000H, 0001H, 0002H, 0003H, and so on.
  Following count up operation, the counter counts down upon a match between the 16-bit count value and the CCR0 buffer register.
- Mode other than above
  The 16-bit counter starts counting from the initial value FFFFH.
  It counts up FFFFH, 0000H, 0001H, 0002H, 0003H, and so on.

(b) Clear operation

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and cleared, when the value of the 16-bit counter is captured and cleared, when the edge of the encoder clear signal is detected and cleared, and when the clear level condition of the TENCm0, TENCm1, and TECRm pins is detected and cleared. The count operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTTEQCn0 and INTTTEQCn1 interrupt signals are not generated.

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running timer mode, pulse width measurement mode, and encoder compare mode. If the counter overflows, the TTmOPT0.TTmOVF bit is set to 1 and an interrupt request signal (INTTTIOVm) is generated in the free-running timer mode and pulse width measurement mode.
If the counter overflows, the TTmOPT1.TTmEOF bit is set to 1 and an interrupt request signal (INTTTIOVm) is generated in the encoder compare mode.
Note that the INTTTIOVm signal is not generated under the following conditions.

- Immediately after a count operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured and cleared to 0000H in the pulse width measurement mode

Caution After the overflow interrupt request signal (INTTTIOVm) has been generated, be sure to check that the overflow flag (TTmOVF, TTmEOF bits) is set to 1.

(d) Count value holding operation

The value of the 16-bit counter is held by the TTmCTL2.TTmECC bit in the encoder compare mode. The value of the 16-bit counter is reset to FFFFH when the TTmECC bit = 0 and TTmCTL0.TTmCE bit = 0. When the TTmCE bit is set to 1 next time, the set value of the TTmTCW register is transferred to the 16-bit counter and the counter continues its count operation.

If the TTmECC bit = 1 and TTmCE bit = 0, the value of the 16-bit counter is held. When the TTmCE bit is set to 1 next time, the counter resumes the count operation from the held value.

(e) Counter read operation during count operation

The value of the 16-bit counter of TMTm can be read by using the TTmCNT register during the count operation. When the TTmCTL0.TTmCE bit = 1, the value of the 16-bit counter can be read by reading the TTmCNT register. If the TTmCNT register is read when the TTmCTL2.TTmECC bit = 0 and TTmCE bit = 0, however, it is 0000H. The held value of the TTmCNT register is read if the register is read when the TTmECC bit = 1 and TTmCE bit = 0.

(f) Underflow operation

The 16-bit counter underflow occurs at the timing when the 16-bit counter value changes from 0000H to FFFFH in the encoder compare mode. When underflow occurs, the TTmOPT1.TTmEUF bit is set to 1 and an interrupt request signal (INTTTIOVm) is generated.

(g) Interrupt operation

TMTm generates the following four types of interrupt request signals.

- INTTTEQCm0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TTmCCR0 register.
- INTTTEQCm1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TTmCCR1 register.
- INTTTIOVm interrupt: This signal functions as an overflow interrupt request signal.
- INTTIECm interrupt: This signal functions as a valid edge detection interrupt request signal of the encoder clear input (TECRm pin).

(2) Basic counter operation of TMT2 and TMT3

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

Remark k = 2, 3

(a) Counter start operation

- In external event count mode
  When the TTkCTL0.TTkCE bit is set from 0 to 1, the 16-bit counter is set to 0000H.
  After that, it counts up to 0001H, 0002H, 0003H, … each time the valid edge of external event count input (TITk0) is detected.
- Triangular-wave PWM mode
  The 16-bit counter starts counting from the initial value FFFFH.
  It counts up FFFFH, 0000H, 0001H, 0002H, 0003H, and so on.
  Following count up operation, the counter counts down upon a match between the 16-bit count value and the CCR0 buffer register.
- Mode other than above
  The 16-bit counter starts counting from the initial value FFFFH.
  It counts up FFFFH, 0000H, 0001H, 0002H, 0003H, and so on.

(b) Clear operation

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and cleared, when the value of the 16-bit counter is captured and cleared. The count operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTTEQCk0 and INTTTEQCk1 interrupt signals are not generated.

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running timer mode and pulse width measurement mode. If the counter overflows, the TTkOPT0.TTkOVF bit is set to 1 and an interrupt request signal (INTTTIOVk) is generated.

Note that the INTTTIOVk signal is not generated under the following conditions.

- Immediately after a count operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured and cleared to 0000H in the pulse width measurement mode

Caution After the overflow interrupt request signal (INTTTIOVk) has been generated, be sure to check that the overflow flag (TTkOVF bit) is set to 1.

(d) Counter read operation during count operation

The value of the 16-bit counter of TMTk can be read by using the TTkCNT register during the count operation. When the TTkCTL0.TTkCE bit = 1, the value of the 16-bit counter can be read by reading the TTkCNT register. If the TTkCNT register is read when the TTkCE bit is 0, the value read from the 16-bit counter will be FFFFH and the value read from the TTkCNT register will be 0000H.

(e) Interrupt operation

TMTk generates the following three types of interrupt request signals.

- INTTTEQCk0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TTkCCR0 register.
- INTTTEQCk1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TTkCCR1 register.
- INTTTIOVk interrupt: This signal functions as an overflow interrupt request signal.

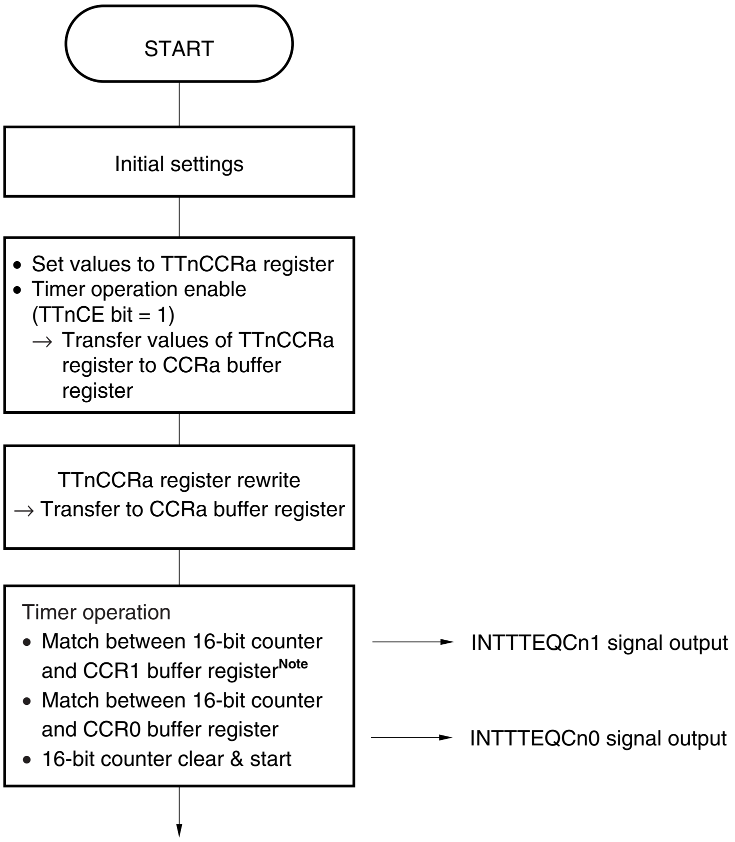
(3) Anytime write and batch write

The TTnCCR0 and TTnCCR1 registers in TMTn can be rewritten during timer operation (TTnCTL0.TTnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 and CCR1 buffer registers differs depending on the mode.

(a) Anytime write

In this mode, data is transferred at any time from the TTnCCR0 and TTnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation (n = 0 to 3).

Figure 8-3. Flowchart of Basic Operation for Anytime Write

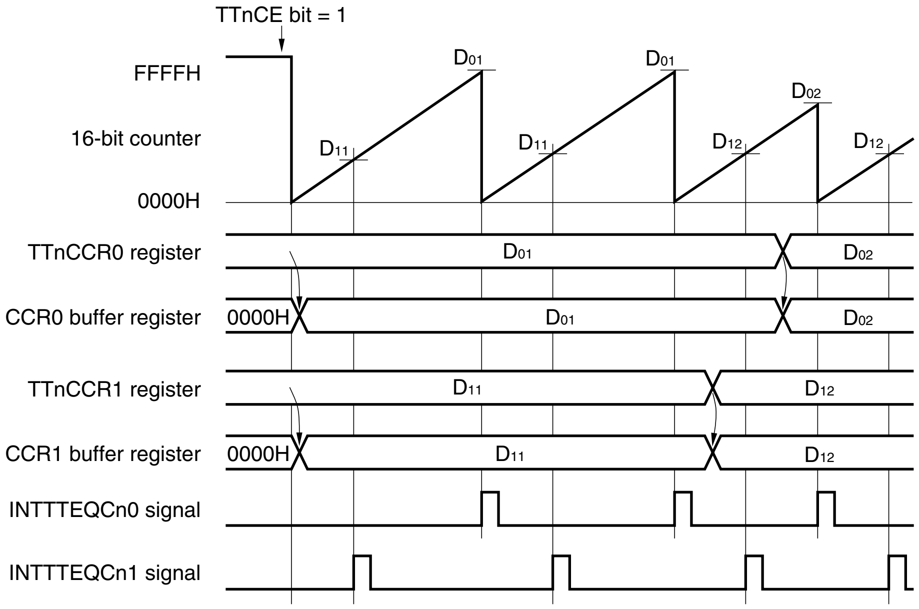

START
Initial settings
• Set values to TTnCCRa register
• Timer operation enable (TTnCE bit = 1)
→ Transfer values of TTnCCRa register to CCRa buffer register
TTnCCRa register rewrite
→ Transfer to CCRa buffer register
Timer operation
• Match between 16-bit counter and CCR1 buffer register^Note
• Match between 16-bit counter and CCR0 buffer register
• 16-bit counter clear & start
INTTTEQCn1 signal output
INTTTEQCn0 signal output

Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Remarks 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0 to 3
a = 0, 1

Figure 8-4. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Set values of TTnCCR0 register

D_{11}, D_{12}: Set values of TTnCCR1 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0 to 3

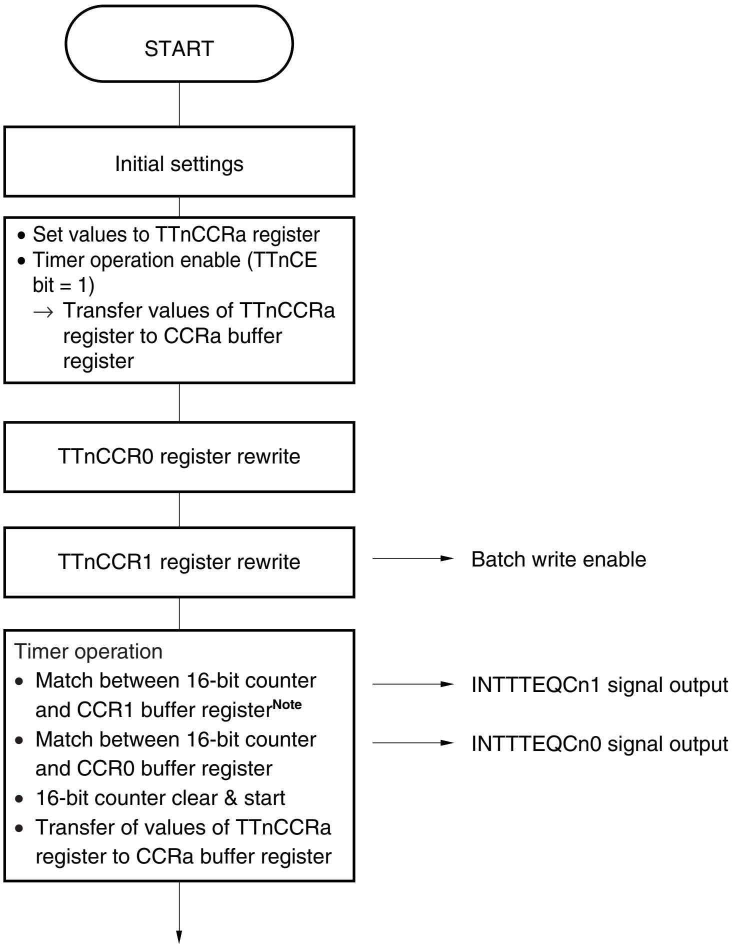
(b) Batch write

In this mode, data is transferred all at once from the TTnCCR0 and TTnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TTnCCR1 register. Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TTnCCR1 register.

In order for the set value when the TTnCCR0 and TTnCCR1 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 and CCR1 buffer registers), it is necessary to rewrite the TTnCCR0 register and then write to the TTnCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. Therefore, the values of the TTnCCR0 and TTnCCR1 registers are transferred to the CCR0 and CCR1 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus even when wishing only to rewrite the value of the TTnCCR0 register, also write the same value (same as preset value of the TTnCCR1 register) to the TTnCCR1 register.

Remark n = 0 to 3

Figure 8-5. Flowchart of Basic Operation for Batch Write


START
Initial settings
• Set values to TTnCCRa register
• Timer operation enable (TTnCE bit = 1)
→ Transfer values of TTnCCRa register to CCRa buffer register
TTnCCR0 register rewrite
TTnCCR1 register rewrite
Batch write enable
Timer operation
• Match between 16-bit counter and CCR1 buffer register Note
• Match between 16-bit counter and CCR0 buffer register
• 16-bit counter clear & start
• Transfer of values of TTnCCRa register to CCRa buffer register
INTTTEQCn1 signal output
INTTTEQCn0 signal output

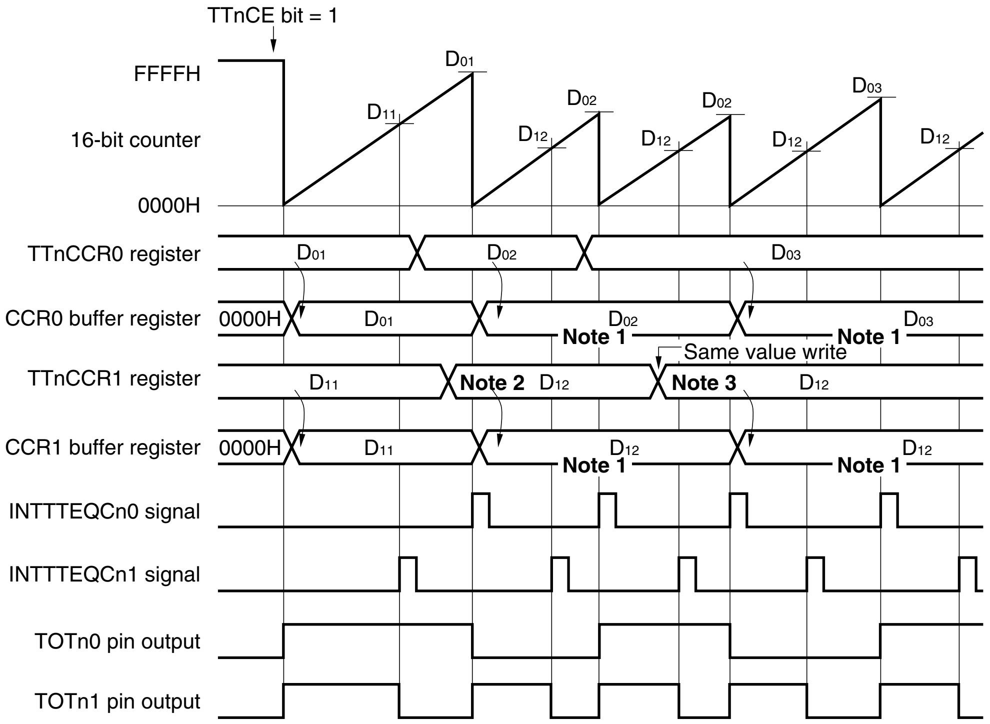
Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Caution Writing to the TTnCCR1 register includes enabling of batch write. Thus, rewrite the TTnCCR1 register after rewriting the TTnCCR0 register.

Remarks 1. The above flowchart illustrates an example of the operation in the PWM output mode.

2. n = 0 to 3
a = 0, 1

Figure 8-6. Timing of Batch Write



Notes 1. Because the TTnCCR1 register was not rewritten, D03 is not transferred.

2. Because the TTnCCR1 register has been written (D12), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TTnCCR0 register (D01).

3. Because the TTnCCR1 register has been written (D12), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TTnCCR0 register (D02).

Remarks 1. D01, D02, D03: Set values of TTnCCR0 register
D11, D12: Set values of TTnCCR1 register

2. The above timing chart illustrates the operation in the PWM output mode as an example.

3. n = 0 to 3

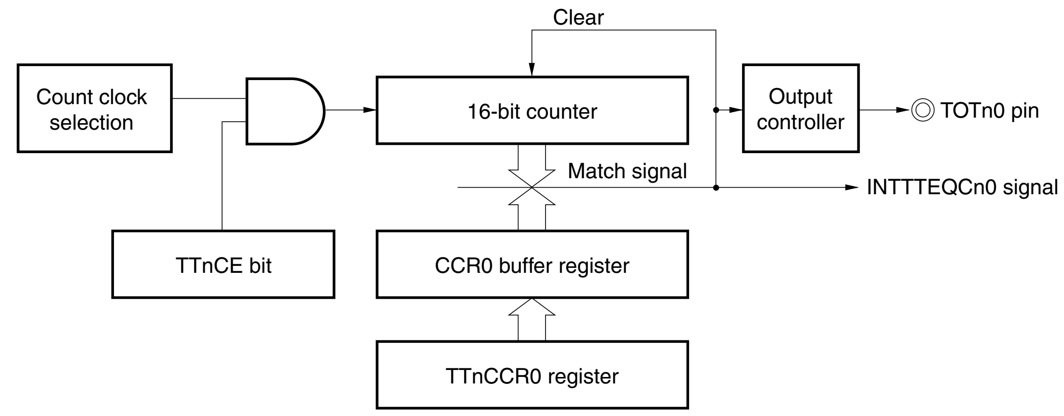
8.6.1 Interval timer mode (TTnMD3 to TTnMD0 bits = 0000)

In the interval timer mode, an interrupt request signal (INTTTEQCn0) is generated at the interval set by the TTnCCR0 register if the TTnCTL0.TTnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOTn0 pin.

The TTnCCR1 register is not used in the interval timer mode. However, the set value of the TTnCCR1 register is transferred to the CCR1 buffer register, and when the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCn1) is generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTTEQCn1 signal is generated, can be output from the TOTn1 pin.

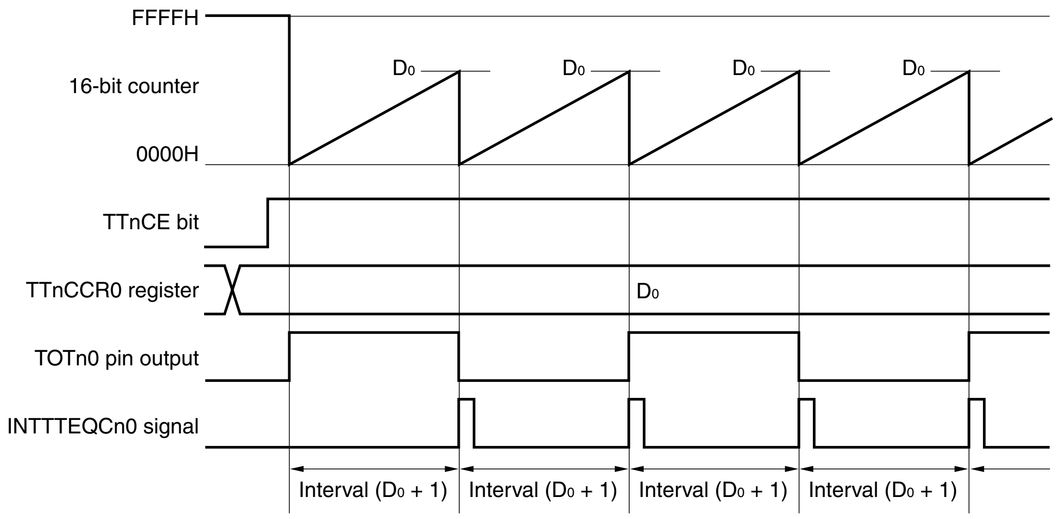
The value of the TTnCCR0 and TTnCCR1 registers can be rewritten even while the timer is operating.

Figure 8-7. Configuration of Interval Timer



Remark n = 0 to 3

Figure 8-8. Basic Timing of Operation in Interval Timer Mode



Remark n = 0 to 3

When the TTnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TOTn0 pin is inverted. Additionally, the set value of the TTnCCR0 register is transferred to the CCR0 buffer register.

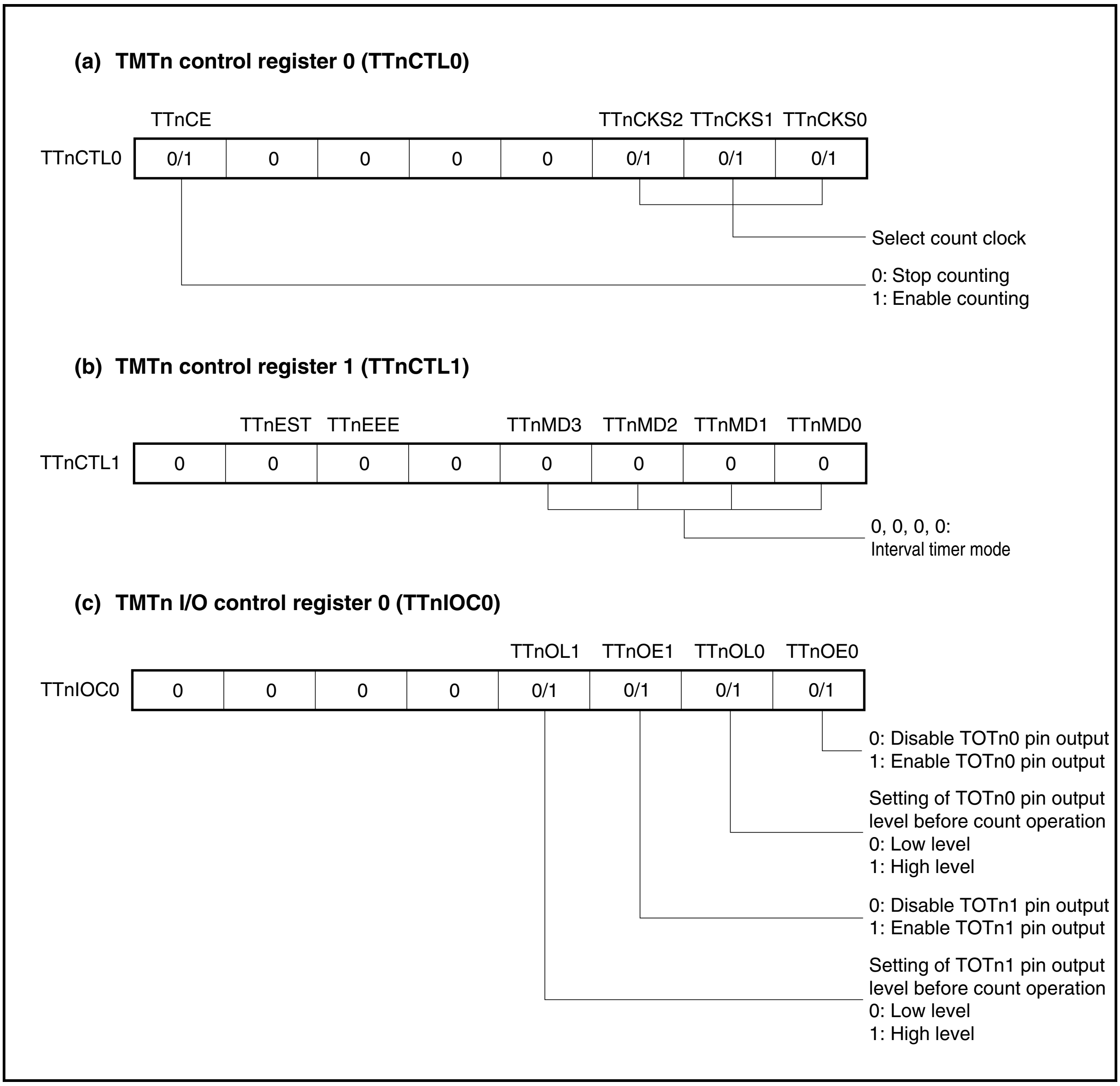
When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TOTn0 pin is inverted, and a compare match interrupt request signal (INTTTEQCn0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TTnCCR0 register + 1) × Count clock cycle

Remark n = 0 to 3

Figure 8-9. Register Setting for Interval Timer Mode Operation (1/2)



(a) TMTn control register 0 (TTnCTL0)

	TTnCE					TTnCKS2	TTnCKS1	TTnCKS0
TTnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TTnCKS2 to TTnCKS0: Select count clock

TTnCE: 0: Stop counting
1: Enable counting

(b) TMTn control register 1 (TTnCTL1)

		TTnEST	TTnEEE		TTnMD3	TTnMD2	TTnMD1	TTnMD0
TTnCTL1	0	0	0	0	0	0	0	0

TTnMD3 to TTnMD0: 0, 0, 0, 0:
Interval timer mode

(c) TMTn I/O control register 0 (TTnIOC0)

					TTnOL1	TTnOE1	TTnOL0	TTnOE0
TTnIOC0	0	0	0	0	0/1	0/1	0/1	0/1

TTnOE0: 0: Disable TOTn0 pin output
1: Enable TOTn0 pin output

TTnOL0: Setting of TOTn0 pin output level before count operation
0: Low level
1: High level

TTnOE1: 0: Disable TOTn1 pin output
1: Enable TOTn1 pin output

TTnOL1: Setting of TOTn1 pin output level before count operation
0: Low level
1: High level

Figure 8-9. Register Setting for Interval Timer Mode Operation (2/2)

(d) TMTn counter read buffer register (TTnCNT)

By reading the TTnCNT register, the count value of the 16-bit counter can be read.

(e) TMTn capture/compare register 0 (TTnCCR0)

If the TTnCCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ Count clock cycle

(f) TMTn capture/compare register 1 (TTnCCR1)

The TTnCCR1 register is not used in the interval timer mode. However, the set value of the TTnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, the TOTn1 pin output is inverted and a compare match interrupt request signal (INTTTEQCn1) is generated.

By setting this register to the same value as the value set in the TTnCCR0 register, a PWM waveform with a duty factor of 50% can be output from the TOTn1 pin.

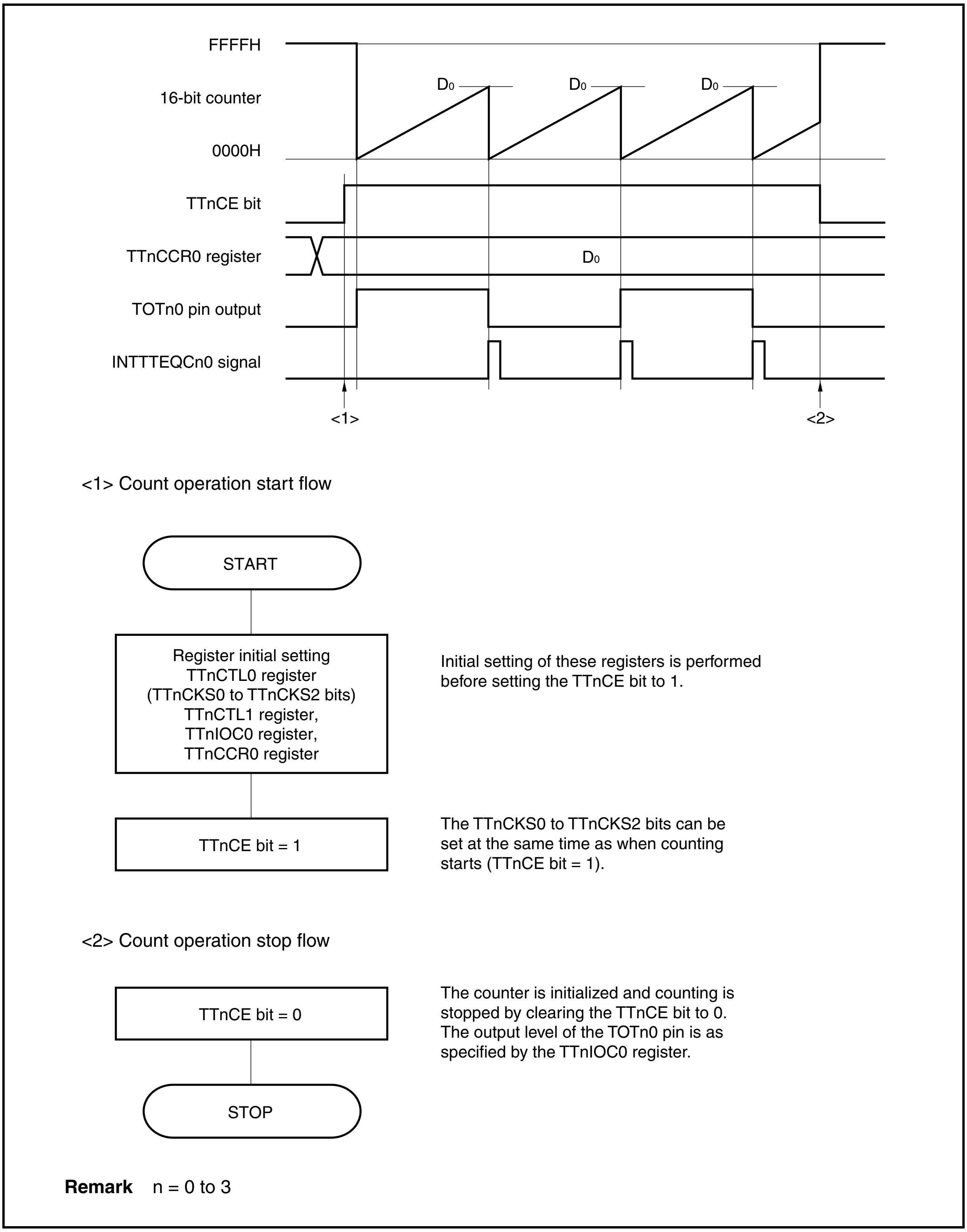
When the TTnCCR1 register is not used, it is recommended to set its value to FFFFH. Also mask the register by the interrupt mask flag (TTnCCIC1.TTnCCMK1).

Remarks 1. TMTm control register 2 (TTmCTL2), TMTn I/O control register 1 (TTnIOC1), TMTn I/O control register 2 (TTnIOC2), TMTm I/O control register 3 (TTmIOC3), TMTn option register 0 (TTnOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the interval timer mode.

2. n = 0 to 3
m = 0, 1

(1) Interval timer mode operation flow

Figure 8-10. Software Processing Flow in Interval Timer Mode



Initial setting of these registers is performed before setting the TTnCE bit to 1.

The TTnCKS0 to TTnCKS2 bits can be set at the same time as when counting starts (TTnCE bit = 1).

The counter is initialized and counting is stopped by clearing the TTnCE bit to 0. The output level of the TOTn0 pin is as specified by the TTnIOC0 register.

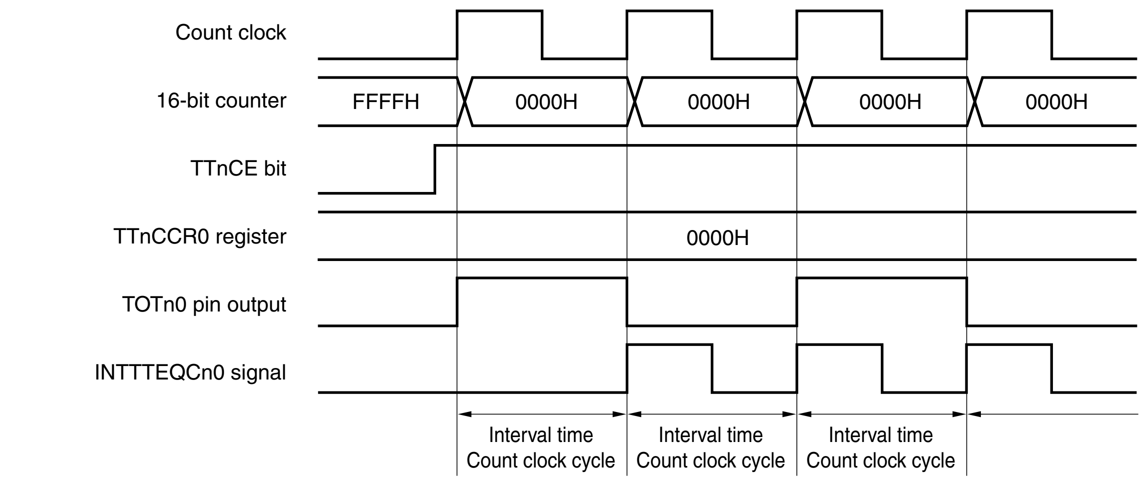
Remark n = 0 to 3

(2) **Interval timer mode operation timing**

(a) **Operation if TTnCCR0 register is set to 0000H**

If the TTnCCR0 register is set to 0000H, the INTTTEQCn0 signal is generated at each count clock, and the output of the TOTn0 pin is inverted.

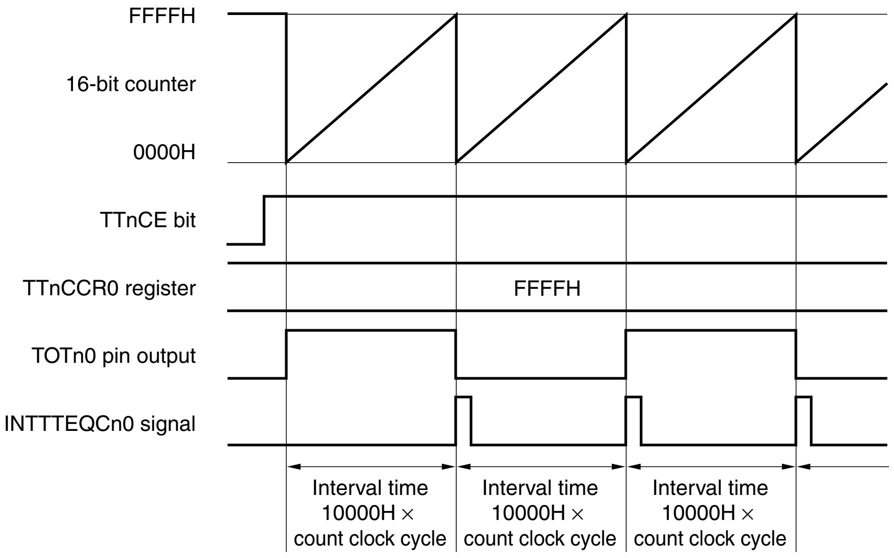
The value of the 16-bit counter is always 0000H.



Remark n = 0 to 3

(b) **Operation if TTnCCR0 register is set to FFFFH**

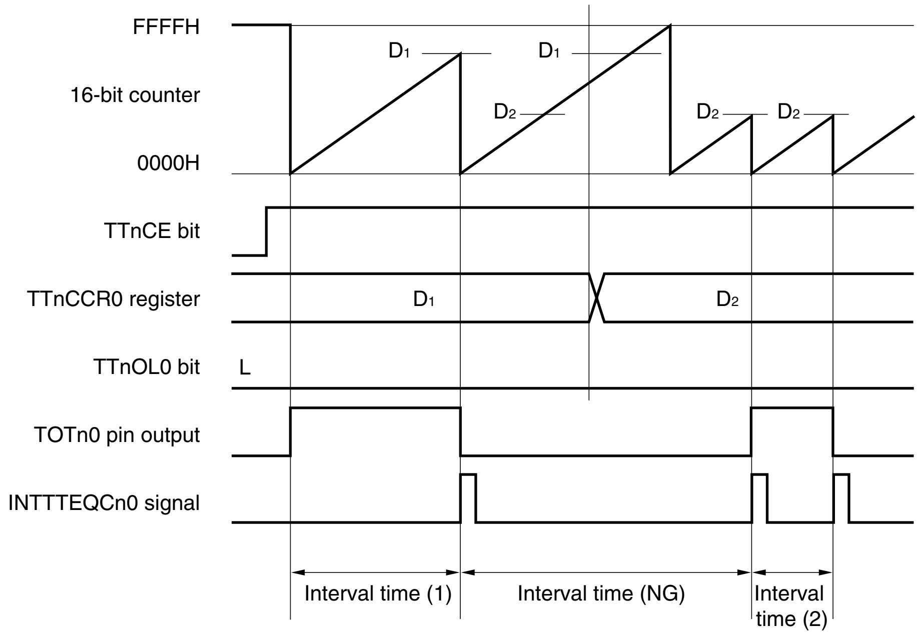
If the TTnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTTEQCn0 signal is generated and the output of the TOTn0 pin is inverted. At this time, an overflow interrupt request signal (INTTTIOVn) is not generated, nor is the overflow flag (TTnOPT0.TTnOVF bit) set to 1.



Remark n = 0 to 3

(c) Notes on rewriting TTnCCR0 register

If the value of the TTnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



Remarks **1.** Interval time (1): $(D_1 + 1)$ × Count clock cycle
Interval time (NG): $(10000H + D_2 + 1)$ × Count clock cycle
Interval time (2): $(D_2 + 1)$ × Count clock cycle

2. n = 0 to 3

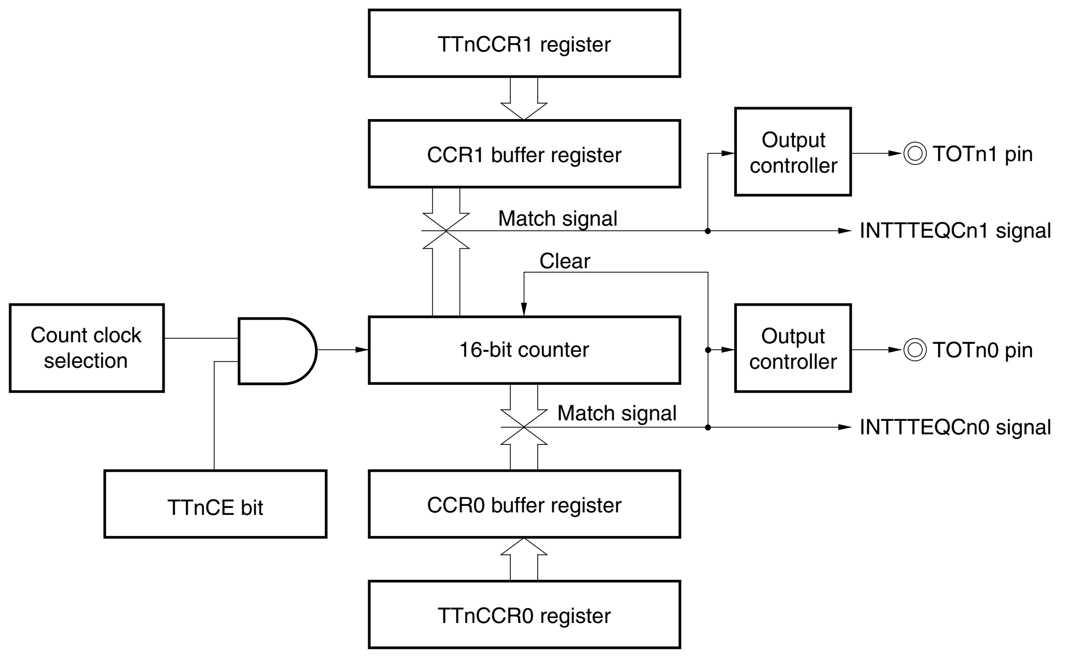
If the value of the TTnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TTnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTTEQCn0 signal is generated and the output of the TOTn0 pin is inverted.

Therefore, the INTTTEQCn0 signal may not be generated at the interval time "$(D_1 + 1)$ × Count clock cycle" or "$(D_2 + 1)$ × Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1)$ × Count clock cycle".

(d) Operation of TTnCCR1 register

Figure 8-11. Configuration of TTnCCR1 Register



Remark n = 0 to 3

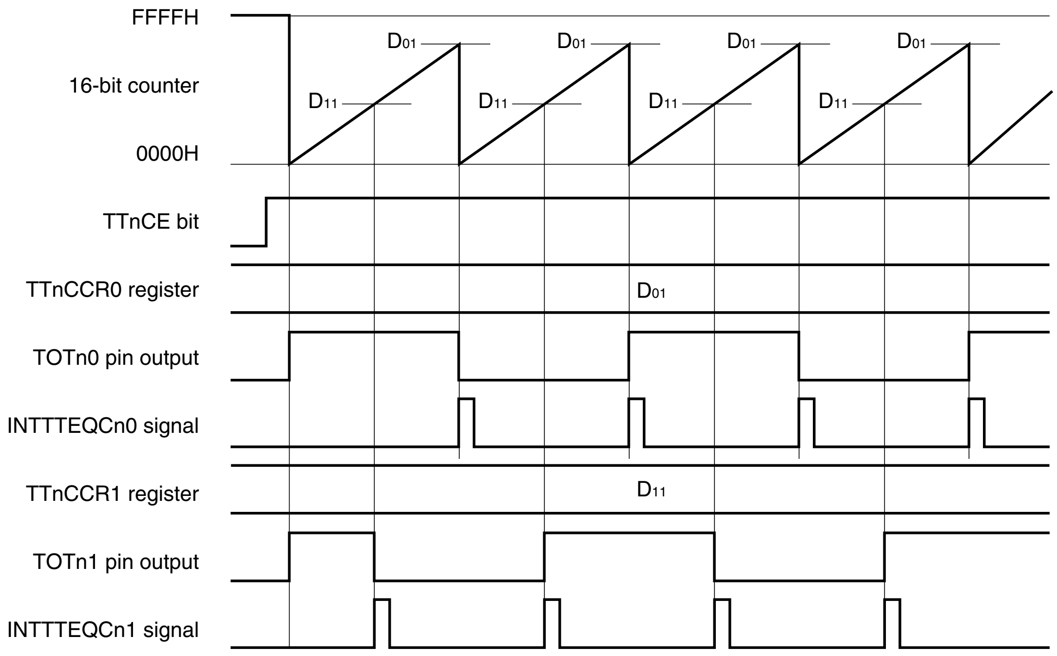
When the TTnCCR1 register is set to the same value as the TTnCCR0 register, the INTTTEQCn0 signal is generated at the same timing as the INTTTEQCn1 signal and the TOTn1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOTn1 pin.

The following shows the operation when the TTnCCR1 register is set to other than the value set in the TTnCCR0 register.

If the set value of the TTnCCR1 register is less than the set value of the TTnCCR0 register, the INTTTEQCn1 signal is generated once per cycle. At the same time, the output of the TOTn1 pin is inverted.

The TOTn1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

Figure 8-12. Timing Chart When $D_{01} \geq D_{11}$

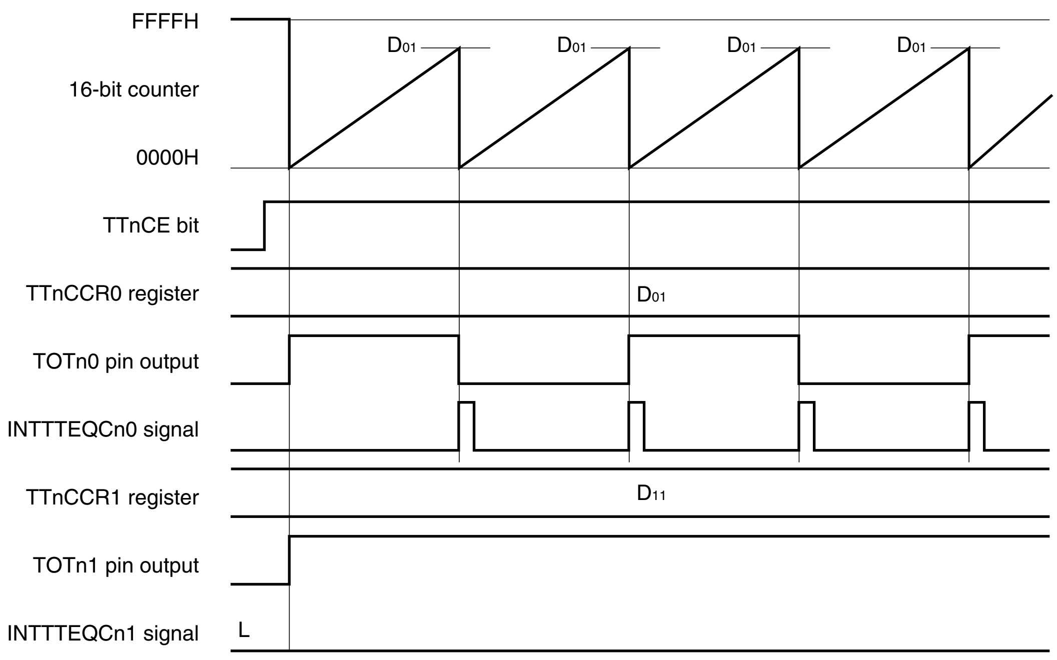


Remark n = 0 to 3

If the set value of the TTnCCR1 register is greater than the set value of the TTnCCR0 register, the count value of the 16-bit counter does not match the value of the TTnCCR1 register. Consequently, the INTTTEQCn1 signal is not generated, nor is the output of the TOTn1 pin changed.

When the TTnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 8-13. Timing Chart When $D_{01} < D_{11}$



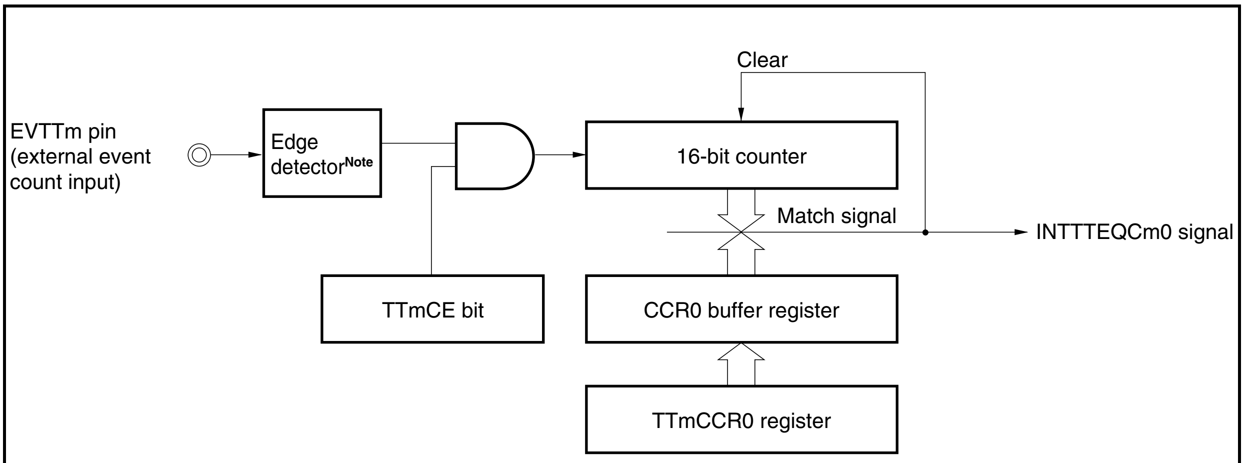
Remark n = 0 to 3

8.6.2 External event count mode (TTnMD3 to TTnMD0 bits = 0001)

In the external event count mode, the valid edge of the external event count input (the EVTTm pin in the case of TMT0 and TMT1, and the TITk0 pin in the case of TMT2 and TMT3) is counted when the TTnCTL0.TTnCE bit is set to 1, and an interrupt request signal (INTTTEQCn0) is generated each time the number of edges set by the TTnCCR0 register have been counted. The TOTn0 and TOTn1 pins cannot be used.

The TTnCCR1 register is not used in the external event count mode.

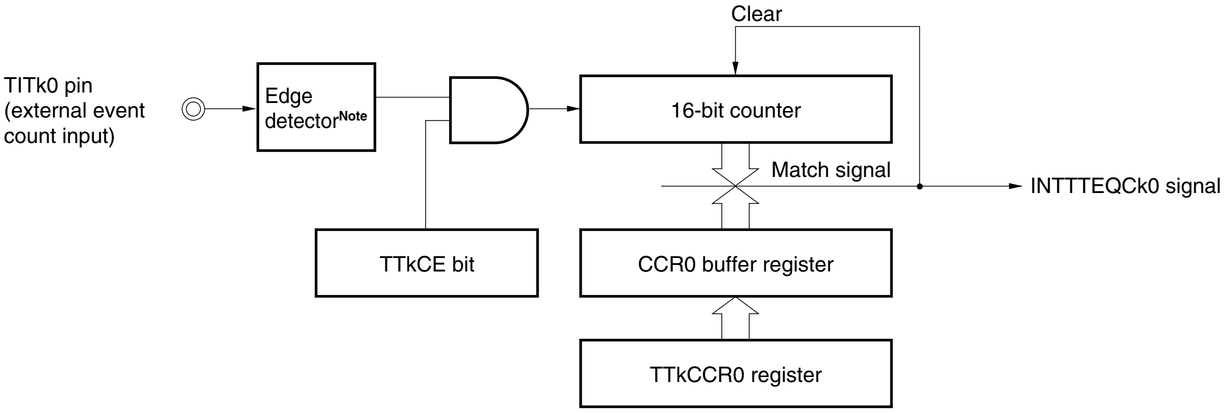
Figure 8-14. Configuration of TMT0 and TMT1 in External Event Count Mode



Note Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark m = 0, 1

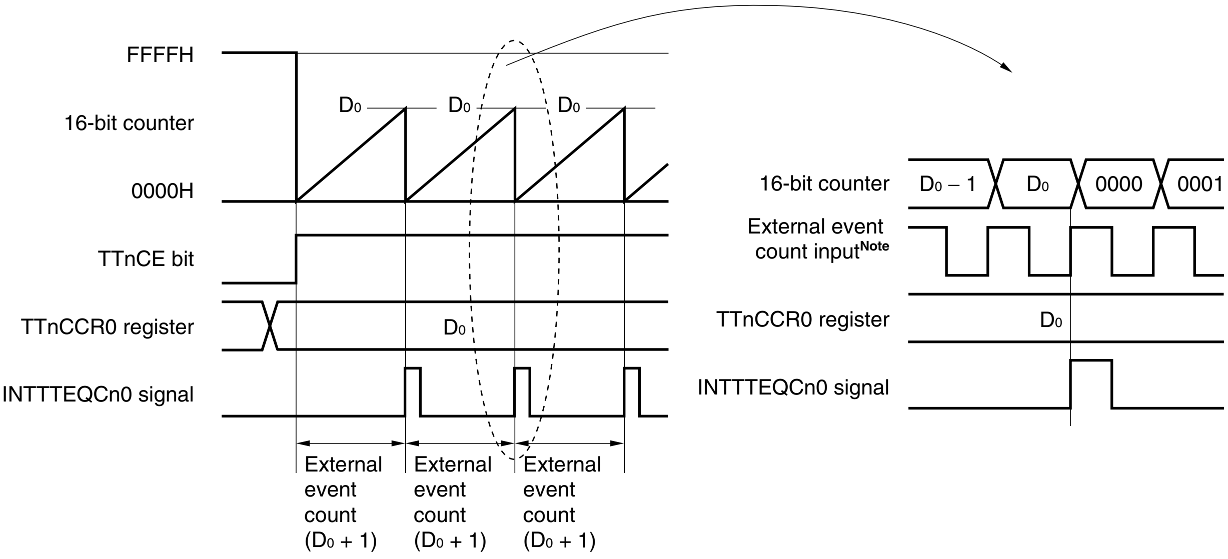
Figure 8-15. Configuration of TMT2 and TMT3 in External Event Count Mode



Note Set by the TTkIOC2.TTkEES1 and TTkIOC2.TTkEES0 bits.

Remark k = 2, 3

Figure 8-16. Basic Timing in External Event Count Mode



Note TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

Remarks 1. This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

2. n = 0 to 3
m = 0, 1
k = 2, 3

When the TTnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TTnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTTEQCn0) is generated.

The INTTTEQCn0 signal is generated each time the valid edge of the external event count input has been detected "value set to TTnCCR0 register + 1" times.

Figure 8-17. Register Setting for Operation in External Event Count Mode (1/2)

(a) TMTn control register 0 (TTnCTL0)

	TTnCE					TTnCKS2	TTnCKS1	TTnCKS0
TTnCTL0	0/1	0	0	0	0	0	0	0

0: Stop counting
1: Enable counting

(b) TMTn control register 1 (TTnCTL1)

		TTnEST	TTnEEE		TTnMD3	TTnMD2	TTnMD1	TTnMD0
TTnCTL1	0	0	0	0	0	0	0	1

0, 0, 0, 1:
External event count mode

(c) TMTn I/O control register 2 (TTnIOC2)

					TTnEES1	TTnEES0	TTnETS1	TTnETS0
TTnIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge of external event count input[Note]

Note TMT0, TMT1: EVTTm pin
TMT2, TMT3: TITk0 pin

(d) TMTn counter read buffer register (TTnCNT)

The count value of the 16-bit counter can be read by reading the TTnCNT register.

(e) TMTn capture/compare register 0 (TTnCCR0)

If the TTnCCR0 register is set to D_0, the count is cleared when the number of external events has reached (D_0 + 1) and the compare match interrupt request signal (INTTTEQCn0) is generated.

Figure 8-17. Register Setting for Operation in External Event Count Mode (2/2)

(f) TMTn capture/compare register 1 (TTnCCR1)

The TTnCCR1 register is not used in the external event count mode. However, the set value of the TTnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCn1) is generated.

When the TTnCCR1 register is not used, it is recommended to set its value to FFFFH. Also mask the register by the interrupt mask flag (TTnCCIC1.TTnCCMK1).

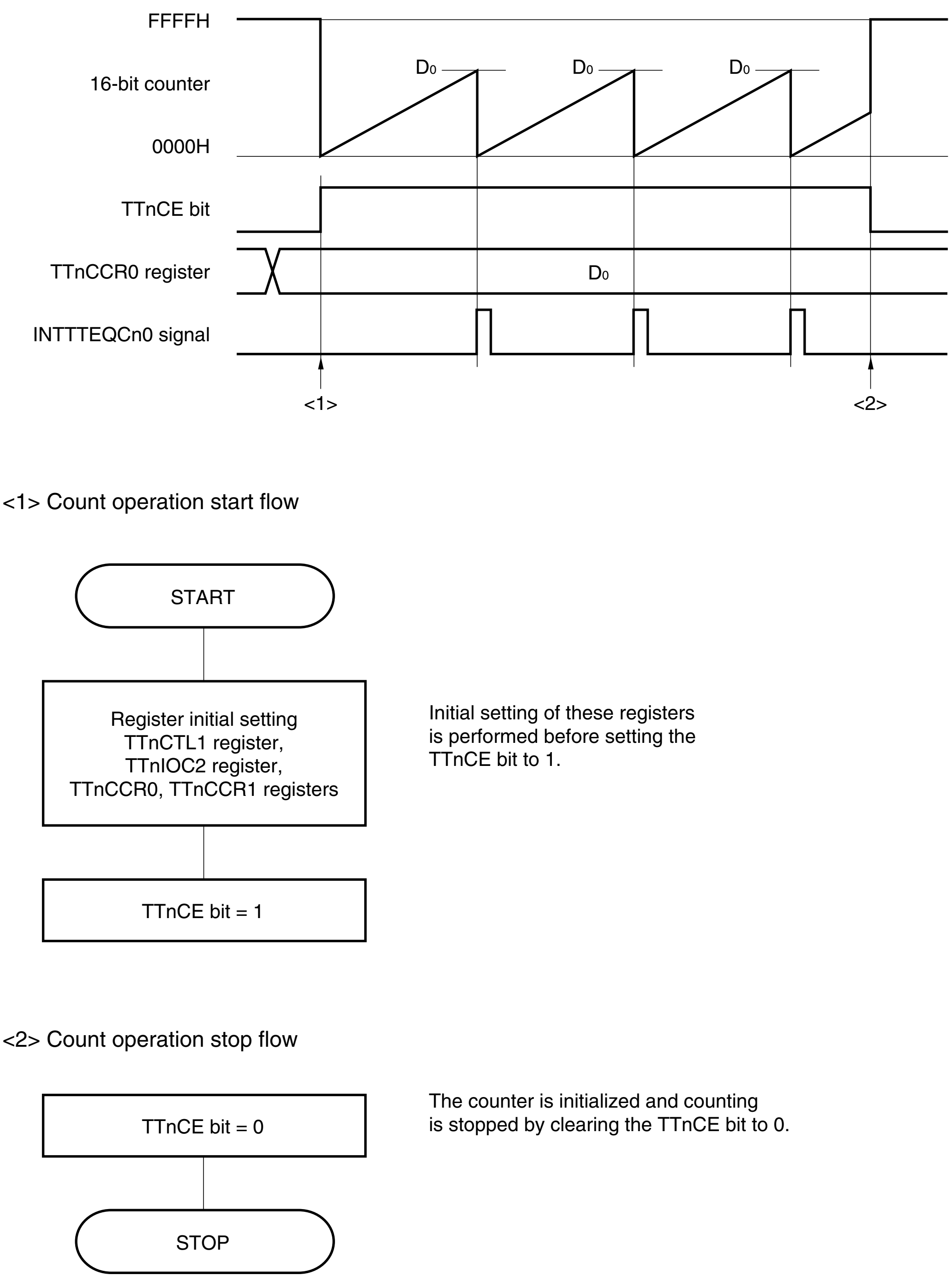
Caution Be sure to set the TTnIOC0 register to 00H.

Remarks 1. TMTm control register 2 (TTmCTL2), TMTn I/O control register 1 (TTnIOC1), TMTm I/O control register 3 (TTmIOC3), TMTn option register 0 (TTnOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the external event count mode.

2. n = 0 to 3
m = 0, 1
k = 2, 3

(1) External event count mode operation flow

Figure 8-18. Software Processing Flow in External Event Count Mode



Initial setting of these registers is performed before setting the TTnCE bit to 1.

The counter is initialized and counting is stopped by clearing the TTnCE bit to 0.

Remark n = 0 to 3

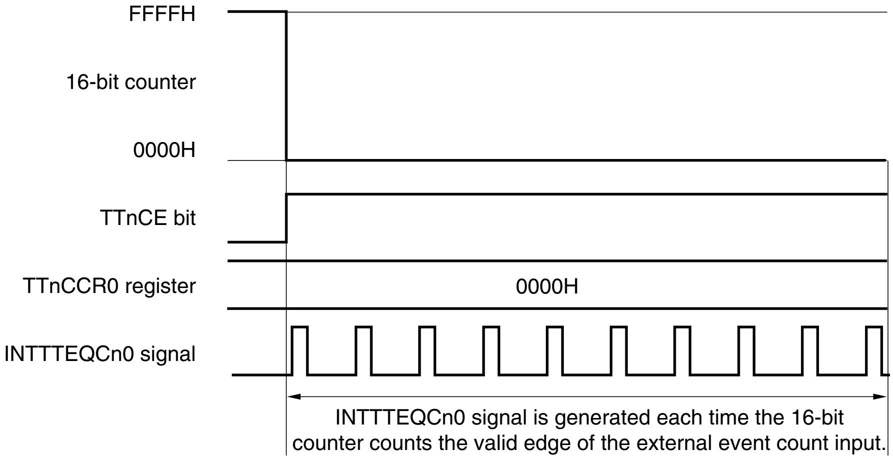
(2) Operation timing in external event count mode

Caution The use of timer output (TOTn0, TOTn1) is prohibited in the external event count mode.

(a) Operation if TTnCCR0 register is set to 0000H

When the TTnCCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates the INTTTEQCn0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

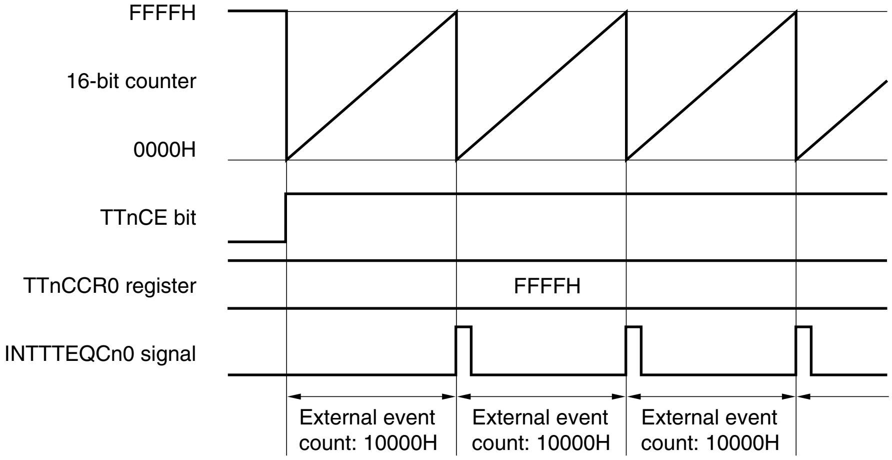
The value of the 16-bit counter is always 0000H.


FFFFH
16-bit counter
0000H
TTnCE bit
TTnCCR0 register
0000H
INTTTEQCn0 signal
INTTTEQCn0 signal is generated each time the 16-bit counter counts the valid edge of the external event count input.

Remark n = 0 to 3

(b) Operation if TTnCCR0 register is set to FFFFH

If the TTnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTTEQCn0 signal is generated. At this time, the TTnOPT0.TTnOVF bit is not set.

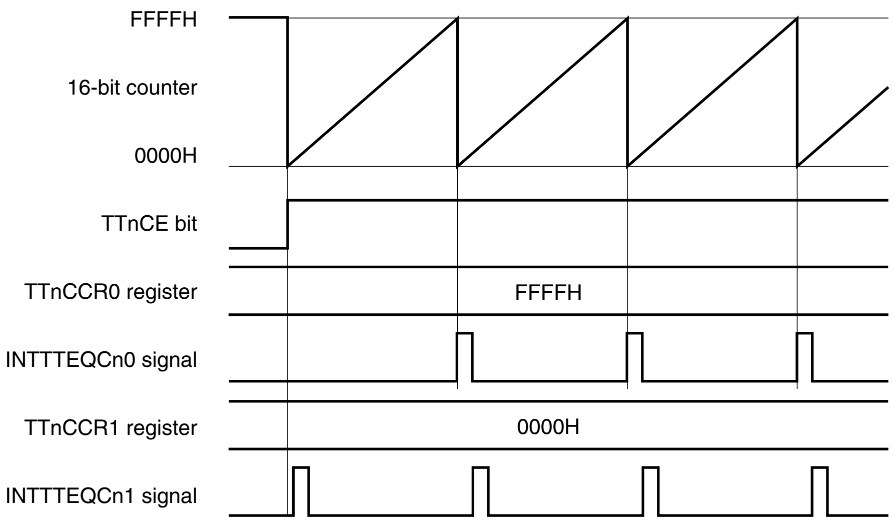

FFFFH
16-bit counter
0000H
TTnCE bit
TTnCCR0 register
FFFFH
INTTTEQCn0 signal
External event count: 10000H
External event count: 10000H
External event count: 10000H

Remark n = 0 to 3

(c) Operation with TTnCCR0 set to FFFFH and TTnCCR1 register to 0000H

When the TTnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTTEQCn0 signal is generated. At this time, the TTnOPT0.TTnOVF bit is not set.

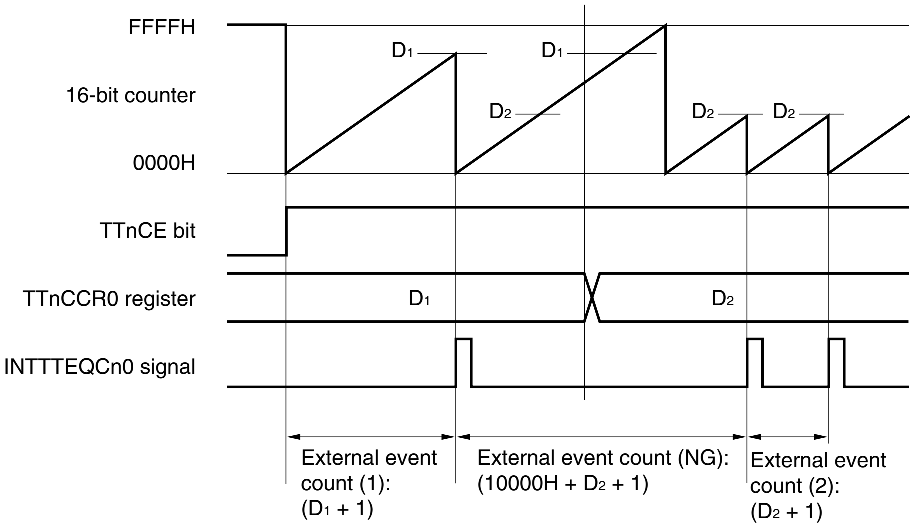
If the TTnCCR1 register is set to 0000H, the INTTTEQCn1 signal is generated when the 16-bit counter is cleared to 0000H.



Remark n = 0 to 3

(d) Notes on rewriting the TTnCCR0 register

If the value of the TTnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.


FFFFH
16-bit counter
0000H
TTnCE bit
TTnCCR0 register
INTTTEQCn0 signal
D1
D1
D2
D2
D2
D1
D2
External event count (1): (D1 + 1)
External event count (NG): (10000H + D2 + 1)
External event count (2): (D2 + 1)

Remark n = 0 to 3

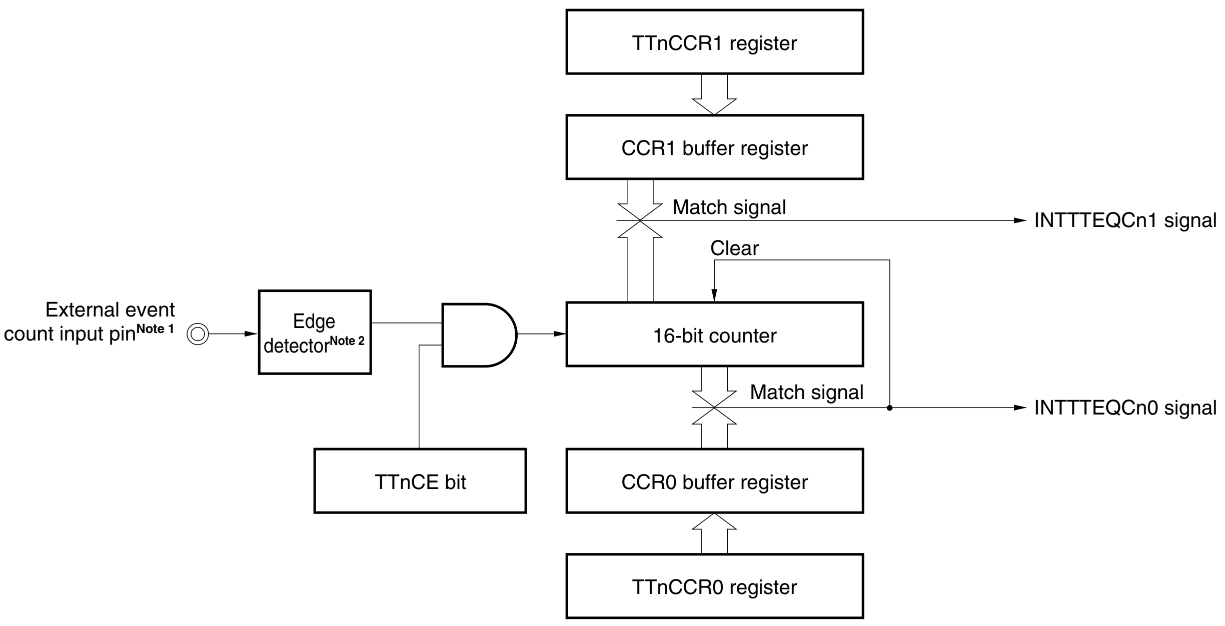
If the value of the TTnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TTnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTTEQCn0 signal is generated.

Therefore, the INTTTEQCn0 signal may not be generated at the valid edge count of “(D_1 + 1) times” or “(D_2 + 1) times” originally expected, but may be generated at the valid edge count of “(10000H + D_2 + 1) times”.

(e) Operation of TTnCCR1 register

Figure 8-19. Configuration of TTnCCR1 Register



Notes 1. TMT0 and TMT1: EVTTm pin input

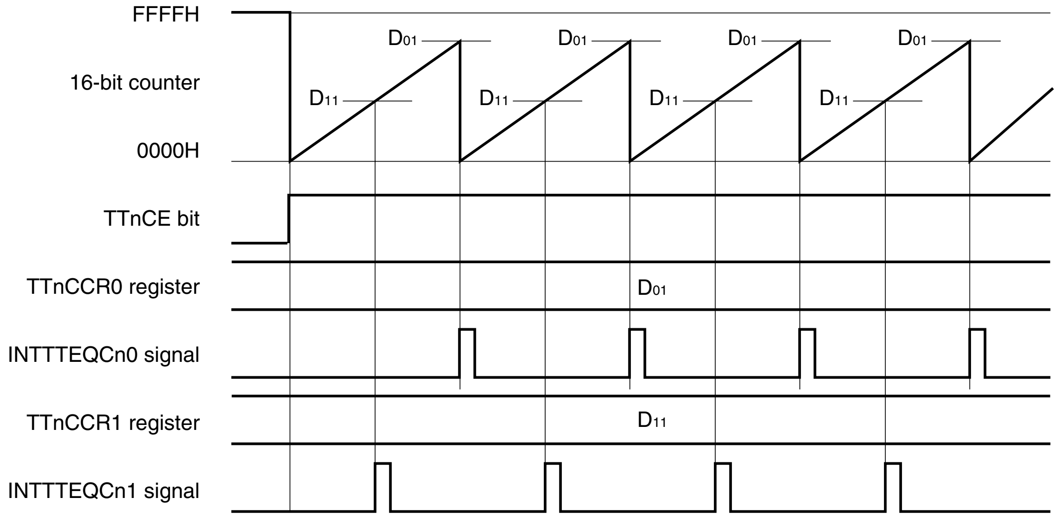
TMT2 and TMT3: TITk0 pin input

2. Set by the TTnIOC2.TTnEES1 and TTnIOC2.TTnEES0 bits.

Remark n = 0 to 3, m = 0, 1, k = 2, 3

If the set value of the TTnCCR1 register is smaller than the set value of the TTnCCR0 register, the INTTTEQCn1 signal is generated once per cycle.

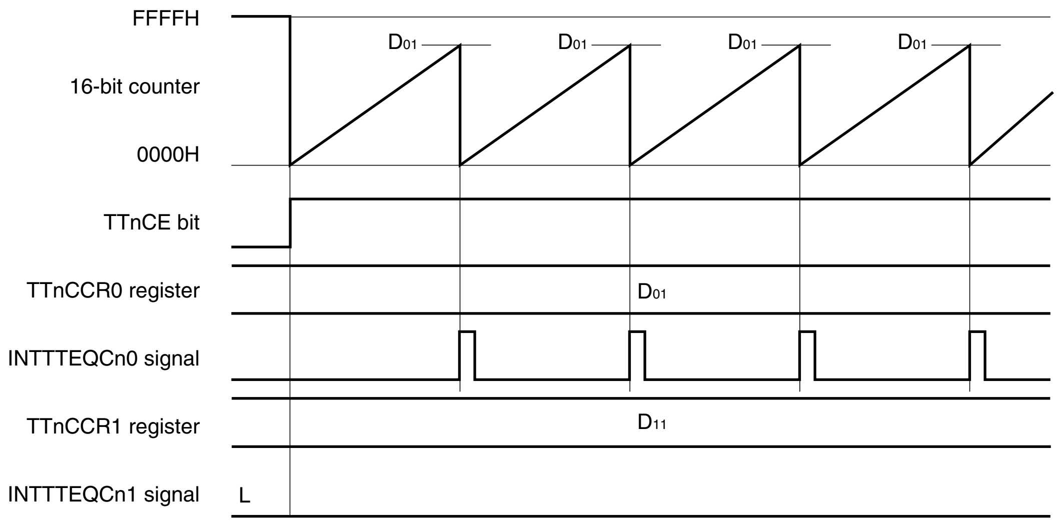
Figure 8-20. Timing Chart When $D_{01} \geq D_{11}$



Remark n = 0 to 3

If the set value of the TTnCCR1 register is greater than the set value of the TTnCCR0 register, the INTTTEQCn1 signal is not generated because the count value of the 16-bit counter and the value of the TTnCCR1 register do not match.
When the TTnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 8-21. Timing Chart When $D_{01} < D_{11}$



Remark n = 0 to 3

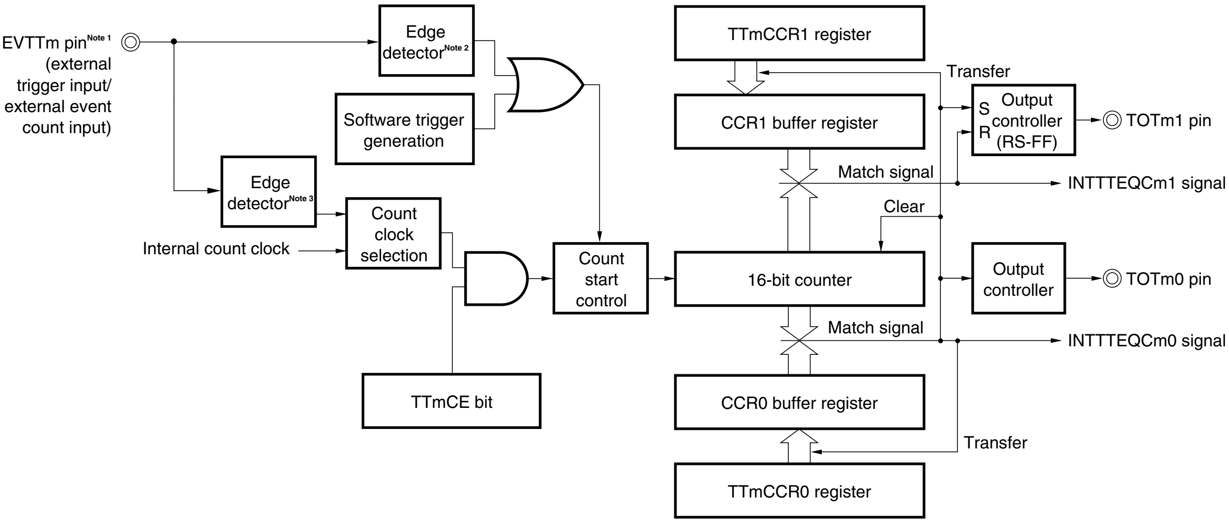
8.6.3 External trigger pulse output mode (TTnMD3 to TTnMD0 bits = 0010)

In the external trigger pulse output mode, 16-bit timer/event counter T waits for a trigger when the TTnCTL0.TTnCE bit is set to 1. When the valid edge of an external trigger input (the EVTTm pin in the case of TMT0 and TMT1, and the TITk0 pin in the case of TMT2 and TMT3) is detected, 16-bit timer/event counter T starts counting, and outputs a PWM waveform from the TOTn1 pin.

For TMT0 and TMT1, a PWM waveform with a duty factor of 50% that has the set value of the TTmCCR0 register + 1 as half its cycle can be output from the TOTm0 pin. Pulses can also be output by generating a software trigger instead of using the external trigger.

For TMT2 and TMT3, pulses can also be output by generating a software trigger instead of using the external trigger input. By using a software trigger, a PWM waveform with a duty factor of 50% that has the set value of the TTkCCR0 register + 1 as half its cycle can be output from the TOTk0 pin.

Figure 8-22. Configuration of TMT0 and TMT1 in External Trigger Pulse Output Mode



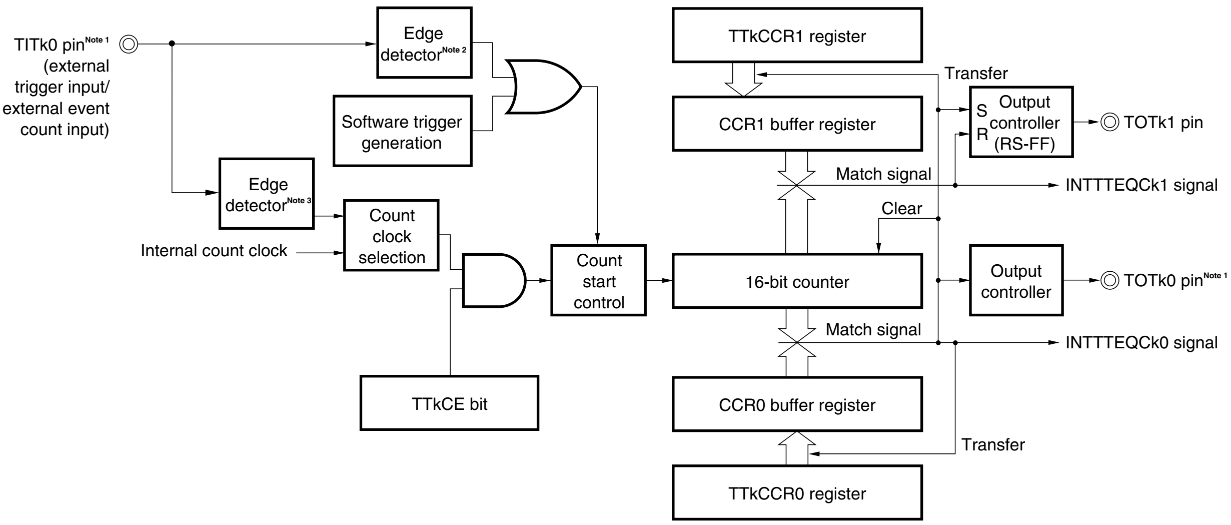
Notes 1. Because the external trigger input pin (EVTTm) and external event count input pin (EVTTm) share the same alternate-function pin, the two functions cannot be used at the same time.

2. Edge detector for external trigger input.
Set by the TTmIOC2.TTmETS1 and TTmIOC2.TTmETS0 bits.

3. Edge detector for external event count input.
Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark m = 0, 1

Figure 8-23. Configuration of TMT2 and TMT3 in External Trigger Pulse Output Mode



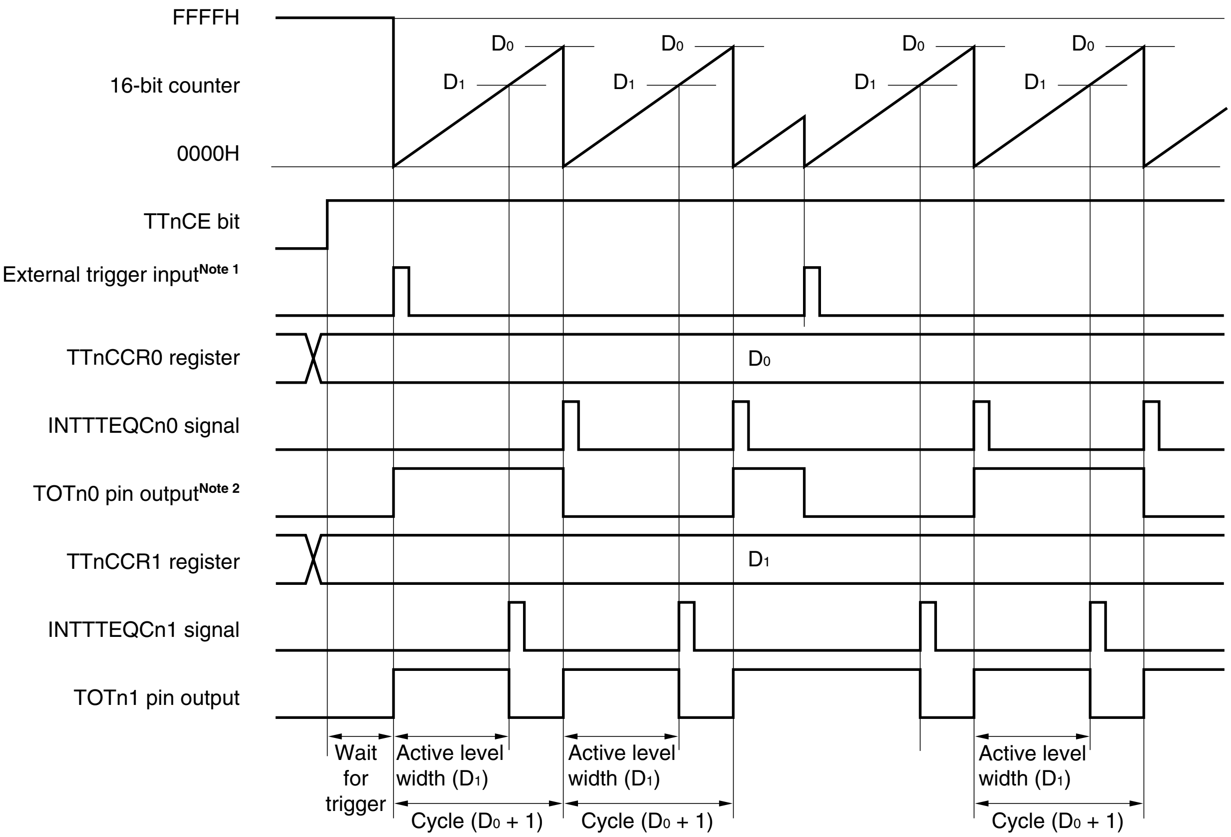
Notes 1. Because the external trigger input pin (TITk0), external event count input pin (TITk0), and timer output pin (TOTk0) share the same pin, the two functions cannot be used at the same time.

2. Edge detector for external trigger input.
Set by the TTkIOC2.TTkETS1 and TTkIOC2.TTkETS0 bits.

3. Edge detector for external event count input.
Set by the TTkIOC2.TTkEES1 and TTkIOC2.TTkEES0 bits.

Remark k = 2, 3

Figure 8-24. Basic Timing in External Trigger Pulse Output Mode



Notes 1. EVTTm pin input in the case of TMT0 and TMT1, and TITk0 pin input in the case of TMT2 and TMT3.

2. In the case of TMT2 and TMT3, this function can only be used by using a software trigger.

16-bit timer/event counter T waits for a trigger when the TTnCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TOTn1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOTn0 pin is inverted. The TOTn1 pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TTnCCR1 register) × Count clock cycle
Cycle = (Set value of TTnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TTnCCR1 register)/(Set value of TTnCCR0 register + 1)

The compare match request signal (INTTTEQCn0) is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal (INTTTEQCn1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TTnCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input (the EVTTm pin in the case of TMT0 and TMT1, and the TITk0 pin in the case of TMT2 and TMT3), or setting the software trigger (TTnCTL1.TTnEST bit) to 1 is used as the trigger.

Remark n = 0 to 3
m = 0, 1
k = 2, 3
a = 0, 1

Figure 8-25. Setting of Registers in External Trigger Pulse Output Mode (1/2)

(a) TMTn control register 0 (TTnTL0)

	TTnCE					TTnCKS2	TTnCKS1	TTnCKS0
TTnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TTnCKS2 to TTnCKS0: Select count clock[Note]

TTnCE:
0: Stop counting
1: Enable counting

Note The setting is invalid when the TTnCTL1.TTnEEE bit = 1.

(b) TMTn control register 1 (TTnCTL1)

		TTnEST	TTnEEE		TTnMD3	TTnMD2	TTnMD1	TTnMD0
TTnCTL1	0	0/1	0/1	0	0	0	1	0

TTnMD3 to TTnMD0:
0, 0, 1, 0:
External trigger pulse output mode

TTnEEE:
0: Operate on count clock selected by TTnCKS0 to TTnCKS2 bits
1: Count with external event count input signal

TTnEST:
Generate software trigger when 1 is written

(c) TMTn I/O control register 0 (TTnIOC0)

					TTnOL1	TTnOE1	TTnOL0	TTnOE0
TTnIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

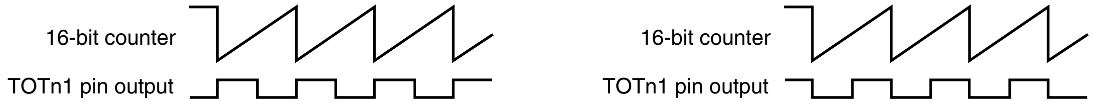
TTnOE0:
0: Disable TOTn0 pin output
1: Enable TOTn0 pin output

TTnOL0:
Setting of TOTn0 pin output level while waiting for external trigger
0: Low level
1: High level

TTnOE1:
0: Disable TOTn1 pin output
1: Enable TOTn1 pin output

TTnOL1:
Setting of TOTn1 pin output level while waiting for external trigger
0: Low level
1: High level

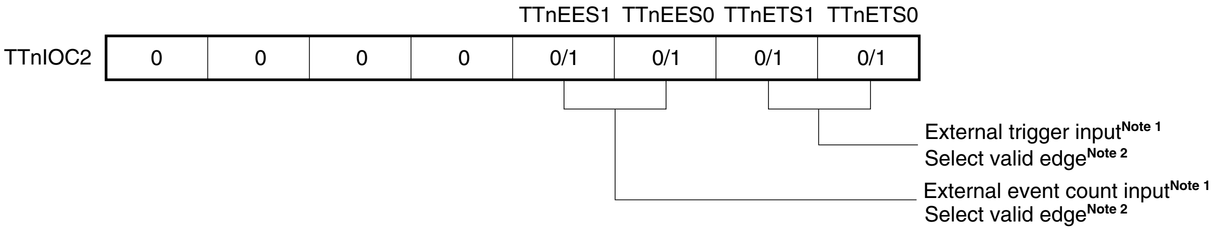
- When TTnOL1 bit = 0
- When TTnOL1 bit = 1



Note Set this bit to 0 when not using the TOTn0 pin in the external trigger pulse output mode.

Figure 8-25. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TMTn I/O control register 2 (TTnIOC2)



					TTnEES1	TTnEES0	TTnETS1	TTnETS0
TTnIOC2	0	0	0	0	0/1	0/1	0/1	0/1

External trigger input[Note 1]
Select valid edge[Note 2]
External event count input[Note 1]
Select valid edge[Note 2]

Notes 1. TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

2. Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(e) TMTn counter read buffer register (TTnCNT)

The value of the 16-bit counter can be read by reading the TTnCNT register.

(f) TMTn capture/compare registers 0 and 1 (TTnCCR0 and TTnCCR1)

If D_0 is set to the TTnCCR0 register and D_1 to the TTnCCR1 register, the cycle and active level of the PWM waveform are as follows.

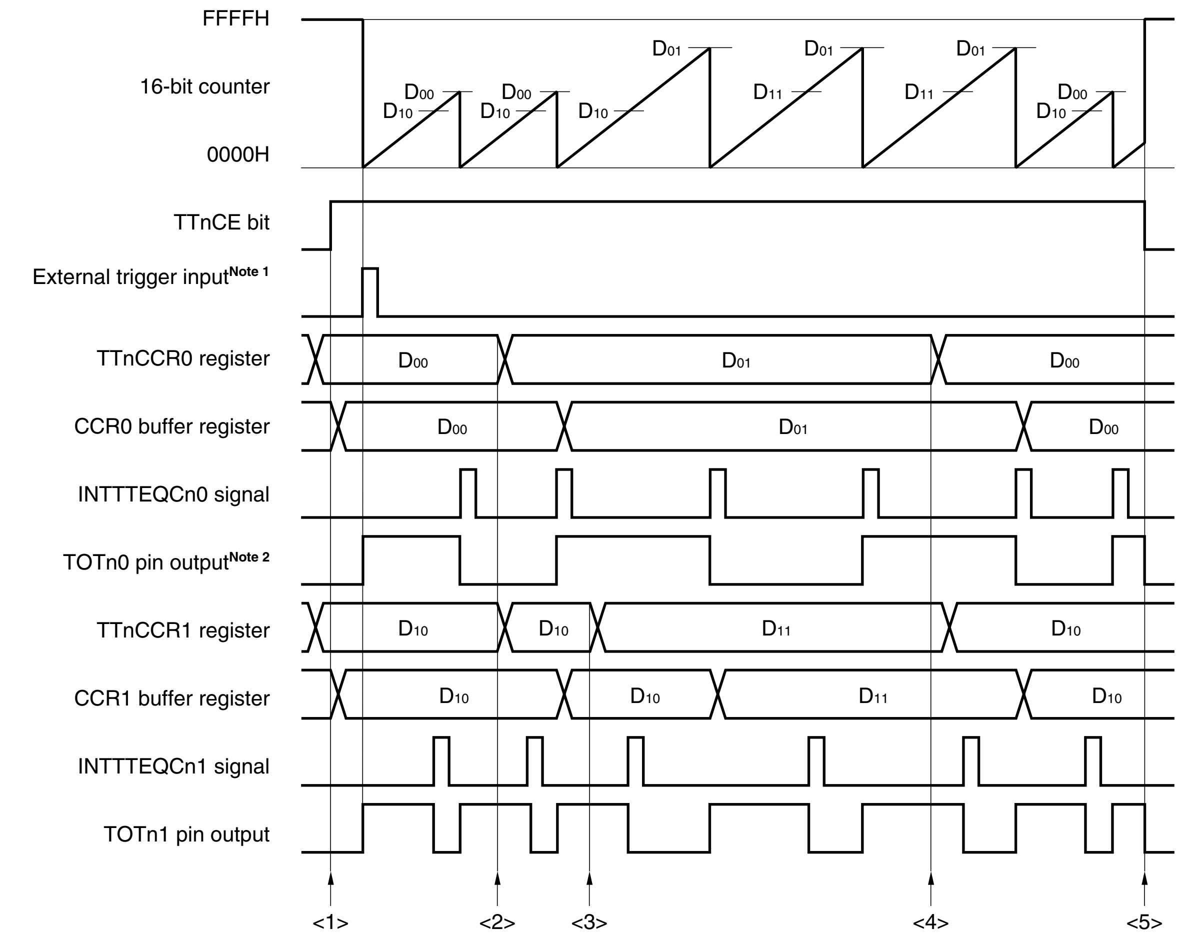
Cycle = (D_0 + 1) × Count clock cycle
Active level width = D_1 × Count clock cycle

Remarks 1. TMTm control register 2 (TTmCTL2), TMTn I/O control register 1 (TTnIOC1), TMTm I/O control register 3 (TTmIOC3), TMTn option register 0 (TTnOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the external trigger pulse output mode.

2. n = 0 to 3
m = 0, 1
k = 2, 3

(1) Operation flow in external trigger pulse output mode

Figure 8-26. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

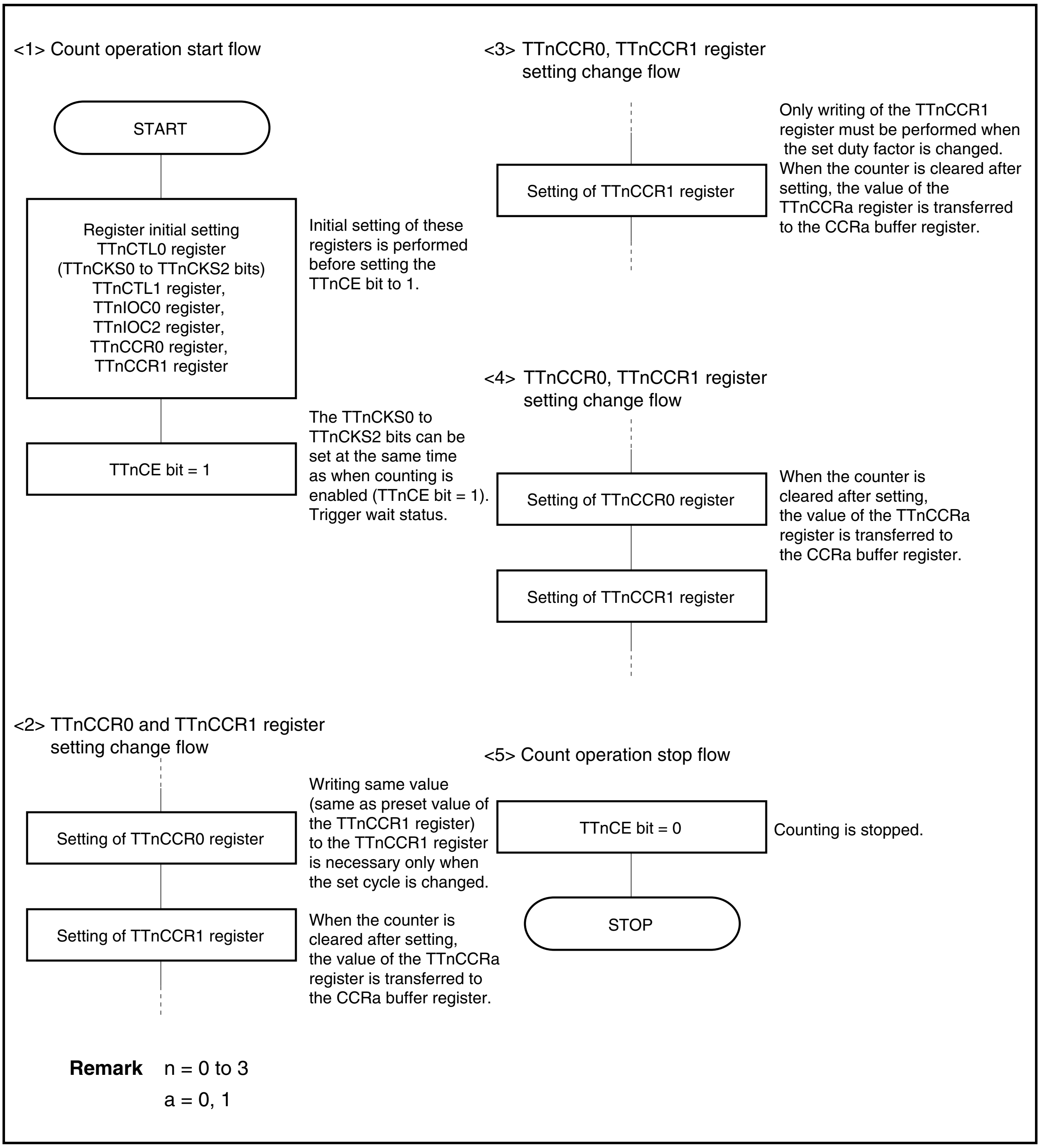


Notes 1. EVTTm pin input in the case of TMT0 and TMT1, and TITk0 pin input in the case of TMT2 and TMT3.

2. In the case of TMT2 and TMT3, this function can only be used by using a software trigger.

Remark n = 0 to 3
m = 0, 1
k = 2, 3

Figure 8-26. Software Processing Flow in External Trigger Pulse Output Mode (2/2)


START
Register initial setting
TTnCTL0 register
(TTnCKS0 to TTnCKS2 bits)
TTnCTL1 register,
TTnIOC0 register,
TTnIOC2 register,
TTnCCR0 register,
TTnCCR1 register
TTnCE bit = 1
Setting of TTnCCR0 register
Setting of TTnCCR1 register
Setting of TTnCCR1 register
Setting of TTnCCR0 register
Setting of TTnCCR1 register
TTnCE bit = 0
STOP

<1> Count operation start flow

Initial setting of these registers is performed before setting the TTnCE bit to 1.

The TTnCKS0 to TTnCKS2 bits can be set at the same time as when counting is enabled (TTnCE bit = 1). Trigger wait status.

<2> TTnCCR0 and TTnCCR1 register setting change flow

Writing same value (same as preset value of the TTnCCR1 register) to the TTnCCR1 register is necessary only when the set cycle is changed.

When the counter is cleared after setting, the value of the TTnCCRa register is transferred to the CCRa buffer register.

<3> TTnCCR0, TTnCCR1 register setting change flow

Only writing of the TTnCCR1 register must be performed when the set duty factor is changed. When the counter is cleared after setting, the value of the TTnCCRa register is transferred to the CCRa buffer register.

<4> TTnCCR0, TTnCCR1 register setting change flow

When the counter is cleared after setting, the value of the TTnCCRa register is transferred to the CCRa buffer register.

<5> Count operation stop flow

Counting is stopped.

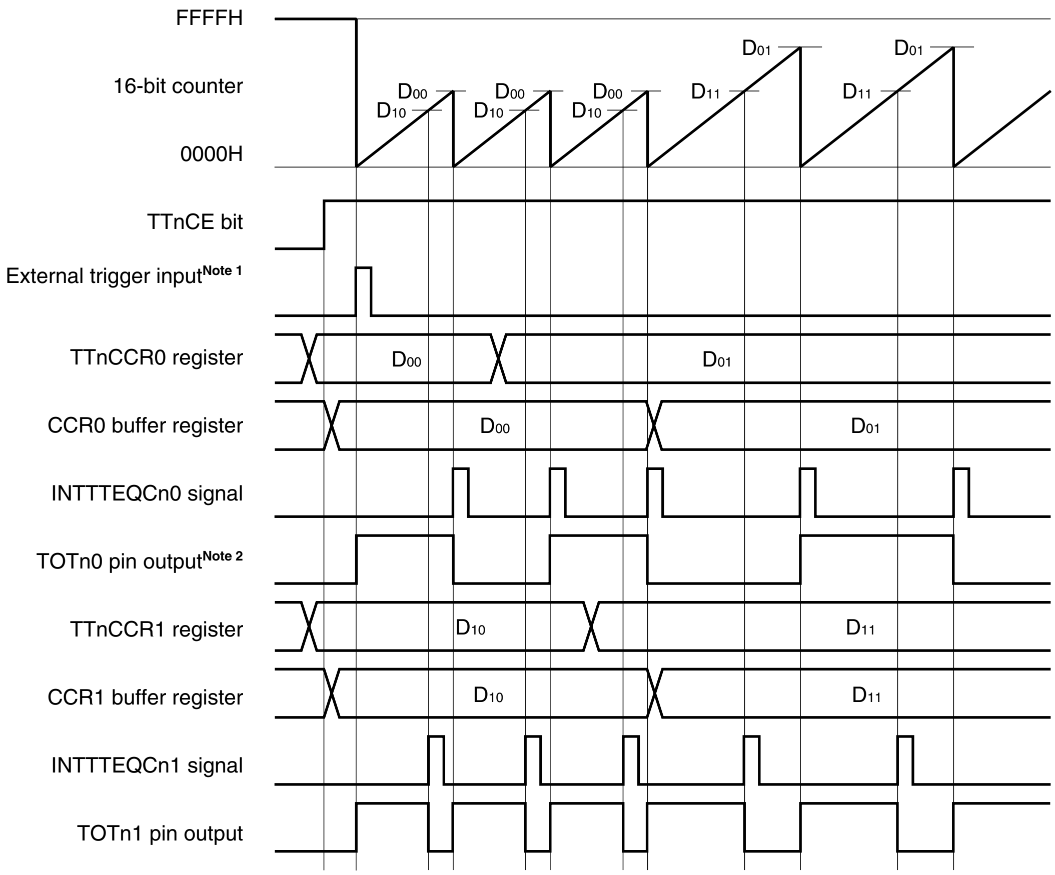
Remark n = 0 to 3
a = 0, 1

(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TTnCCR1 register last.

Rewrite the TTnCCRa register after writing the TTnCCR1 register after the INTTTEQCn0 signal is detected.



Notes 1. TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

2. In the case of TMT2 and TMT3, this function can only be used by using a software trigger.

Remark n = 0 to 3
m = 0, 1
k = 2, 3

In order to transfer data from the TTnCCRa register to the CCRa buffer register, the TTnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TTnCCR0 register and then set the active level width to the TTnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TTnCCR0 register, and then write the same value (same as preset value of the TTnCCR1 register) to the TTnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TTnCCR1 register has to be set.

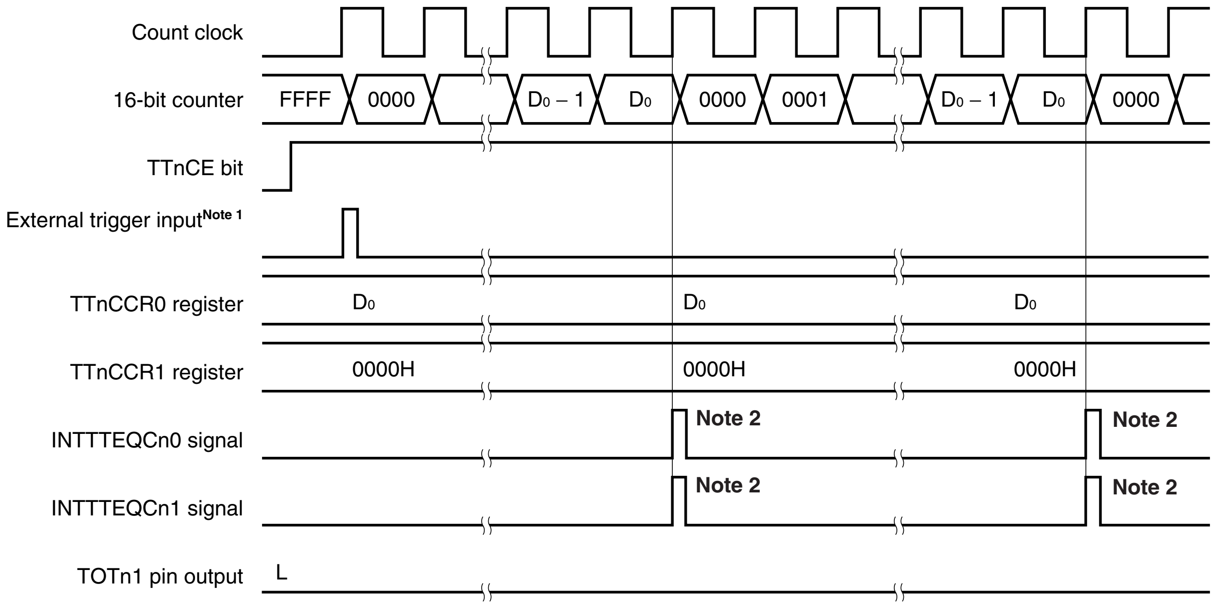
After data is written to the TTnCCR1 register, the value written to the TTnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TTnCCR0 or TTnCCR1 register again after writing the TTnCCR1 register once, do so after the INTTTEQCn0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TTnCCRa register to the CCRa buffer register conflicts with writing the TTnCCRa register.

Remark n = 0 to 3
a = 0, 1

(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TTnCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTTEQCn0 and INTTTEQCn1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.

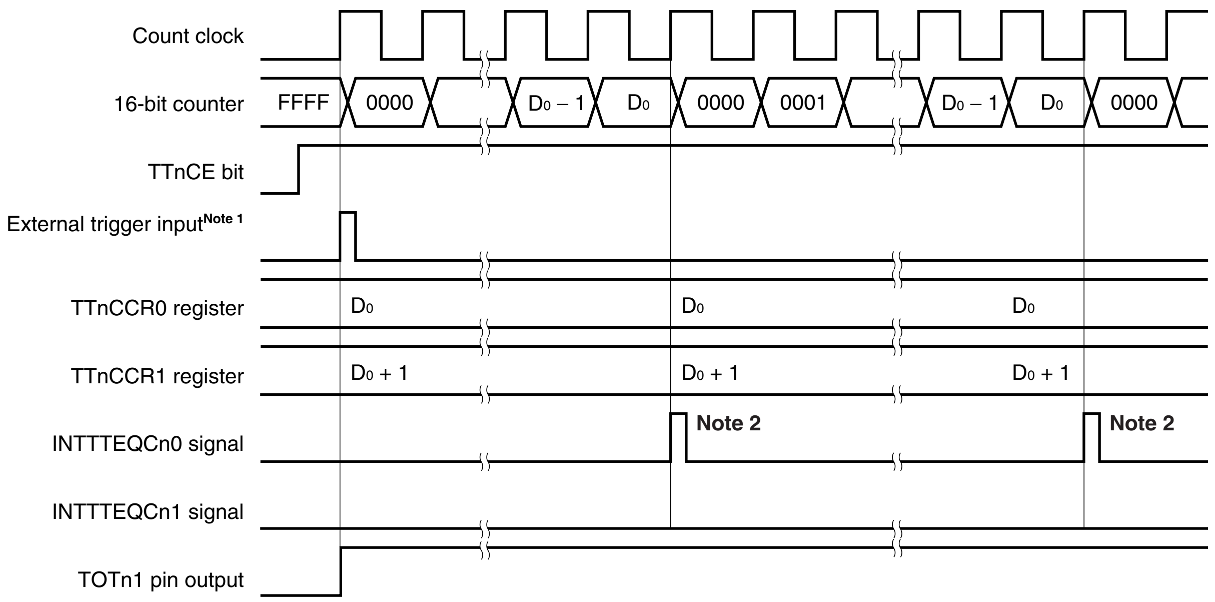


Notes 1. TMT0, TMT1: EVTTm pin input
TMT2, TMT3: TITk0 pin input

2. The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 3
m = 0, 1
k = 2, 3

To output a 100% waveform, set a value of (set value of TTnCCR0 register + 1) to the TTnCCR1 register. If the set value of the TTnCCR0 register is FFFFH, 100% output cannot be produced.



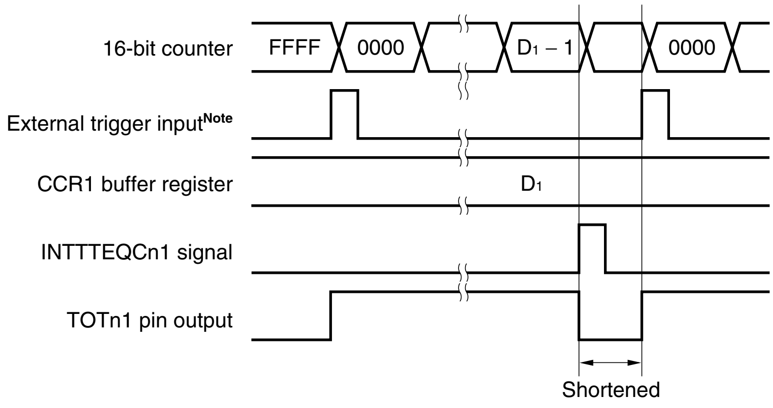
Notes 1. TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

2. The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 3
m = 0, 1
k = 2, 3

(c) Conflict between trigger detection and match with CCR1 buffer register

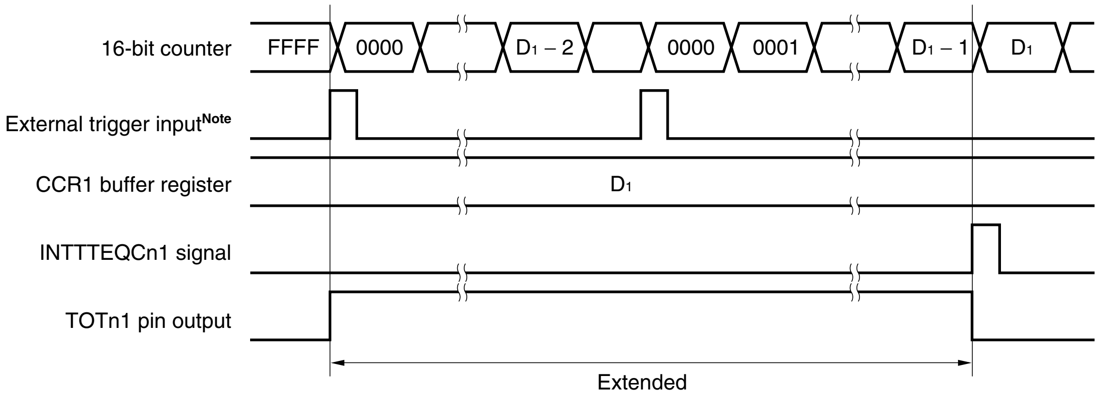
If the trigger is detected immediately after the INTTTEQCn1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOTn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Note TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

Remark n = 0 to 3
m = 0, 1
k = 2, 3

If the trigger is detected immediately before the INTTTEQCn1 signal is generated, the INTTTEQCn1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOTn1 pin remains active. Consequently, the active period of the PWM waveform is extended.

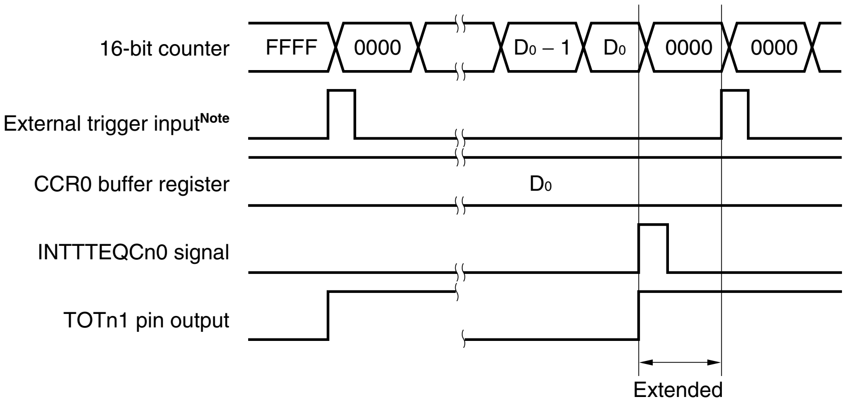


Note TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

Remark n = 0 to 3
m = 0, 1
k = 2, 3

(d) Conflict between trigger detection and match with CCR0 buffer register

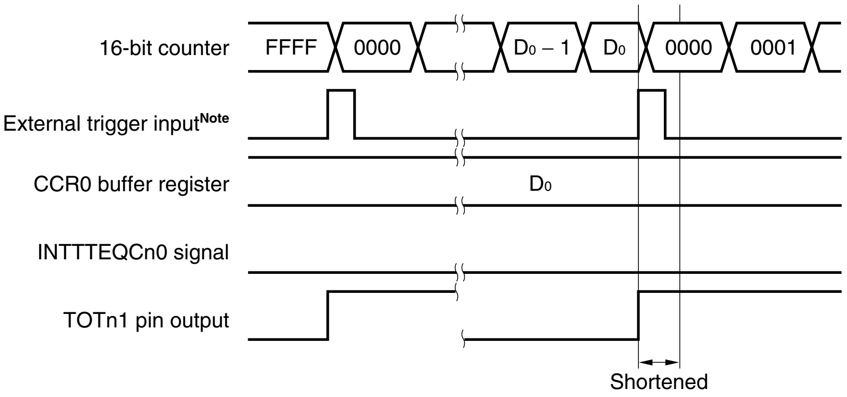
If the trigger is detected immediately after the INTTTEQCn0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOTn1 pin is extended by time from generation of the INTTTEQCn0 signal to trigger detection.



Note TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

Remark n = 0 to 3
m = 0, 1
k = 2, 3

If the trigger is detected immediately before the INTTTEQCn0 signal is generated, the INTTTEQCn0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOTn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

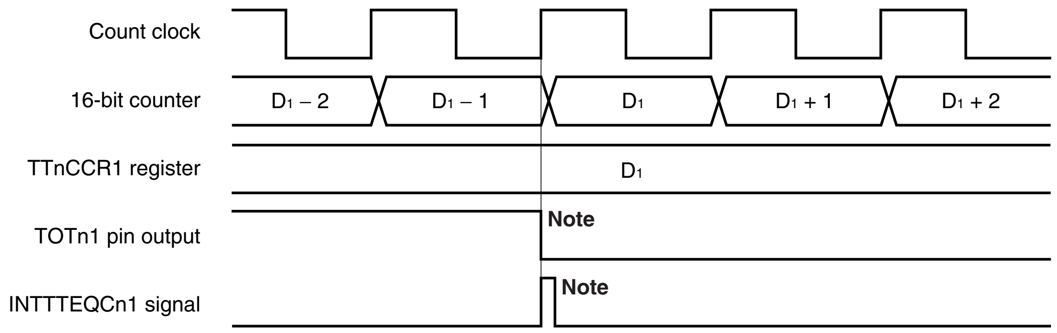


Note TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

Remark n = 0 to 3
m = 0, 1
k = 2, 3

(e) Generation timing of compare match interrupt request signal (INTTTEQCn1)

The timing of generation of the INTTTEQCn1 signal in the external trigger pulse output mode differs from the timing of INTTTEQCn1 signals in other mode; the INTTTEQCn1 signal is generated when the count value of the 16-bit counter matches the value of the TTnCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 3

Usually, the INTTTEQCn1 signal is generated in synchronization with the next count-up, after the count value of the 16-bit counter matches the value of the TTnCCR1 register.

In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOTn1 pin.

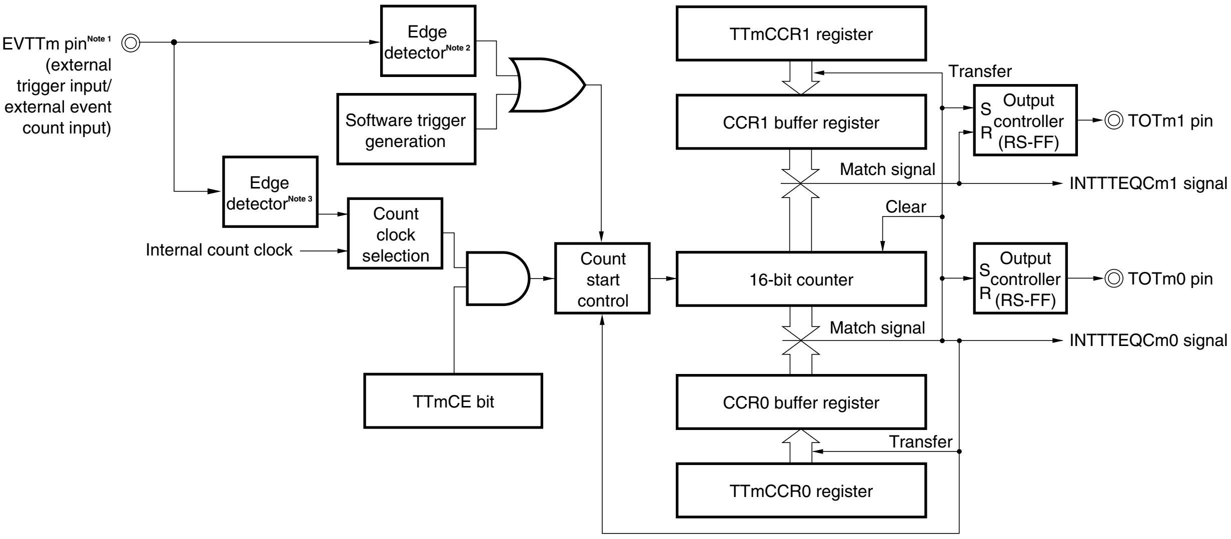
8.6.4 One-shot pulse output mode (TTnMD3 to TTnMD0 bits = 0011)

In the one-shot pulse output mode, 16-bit timer/event counter T waits for a trigger when the TTnCTL0.TTnCE bit is set to 1. When the valid edge of an external trigger input (the EVTTm pin in the case of TMT0 and TMT1, and the TITk0 pin in the case of TMT2 and TMT3) is detected, 16-bit timer/event counter T starts counting, and outputs a one-shot pulse from the TOTn1 pin.

In the case of TMT0 and TMT1, the TOTm0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger). Instead of the external trigger input, a software trigger can also be generated to output the pulse.

In the case of TMT2 and TMT3, instead of the external trigger input, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOTk0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

Figure 8-27. Configuration of TMT0 and TMT1 in One-Shot Pulse Output Mode



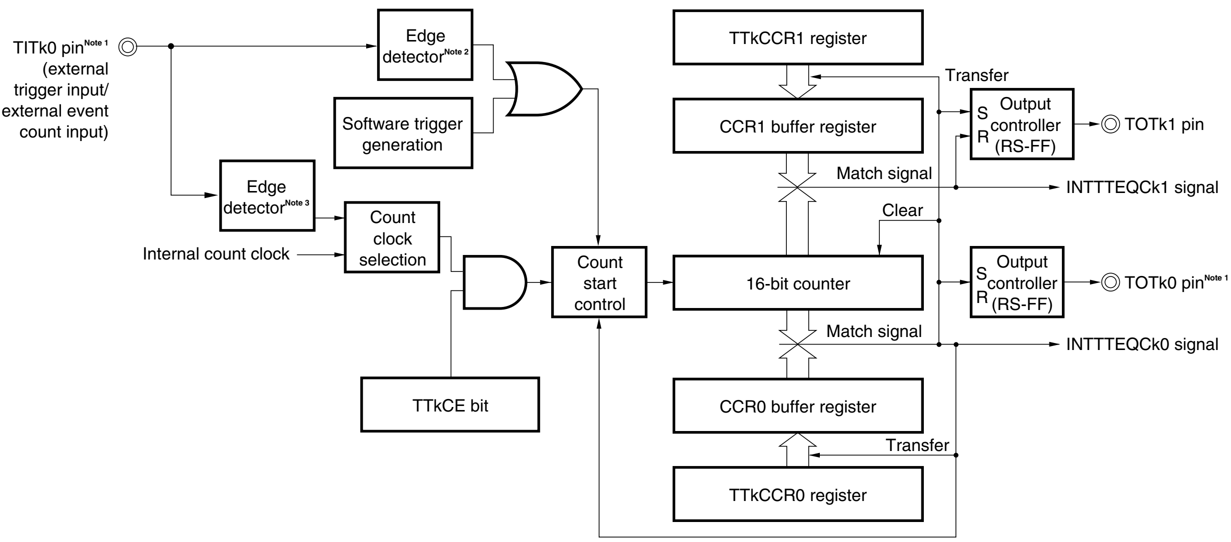
Notes 1. Because the external trigger input pin (EVTTm) and external event count input pin (EVTTm) share the same alternate-function pin, the two functions cannot be used at the same time.

2. Edge detector for external trigger input.
Set by the TTmIOC2.TTmETS1 and TTmIOC2.TTmETS0 bits.

3. Edge detector for external event count input.
Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark m = 0, 1

Figure 8-28. Configuration of TMT2 and TMT3 in One-Shot Pulse Output Mode



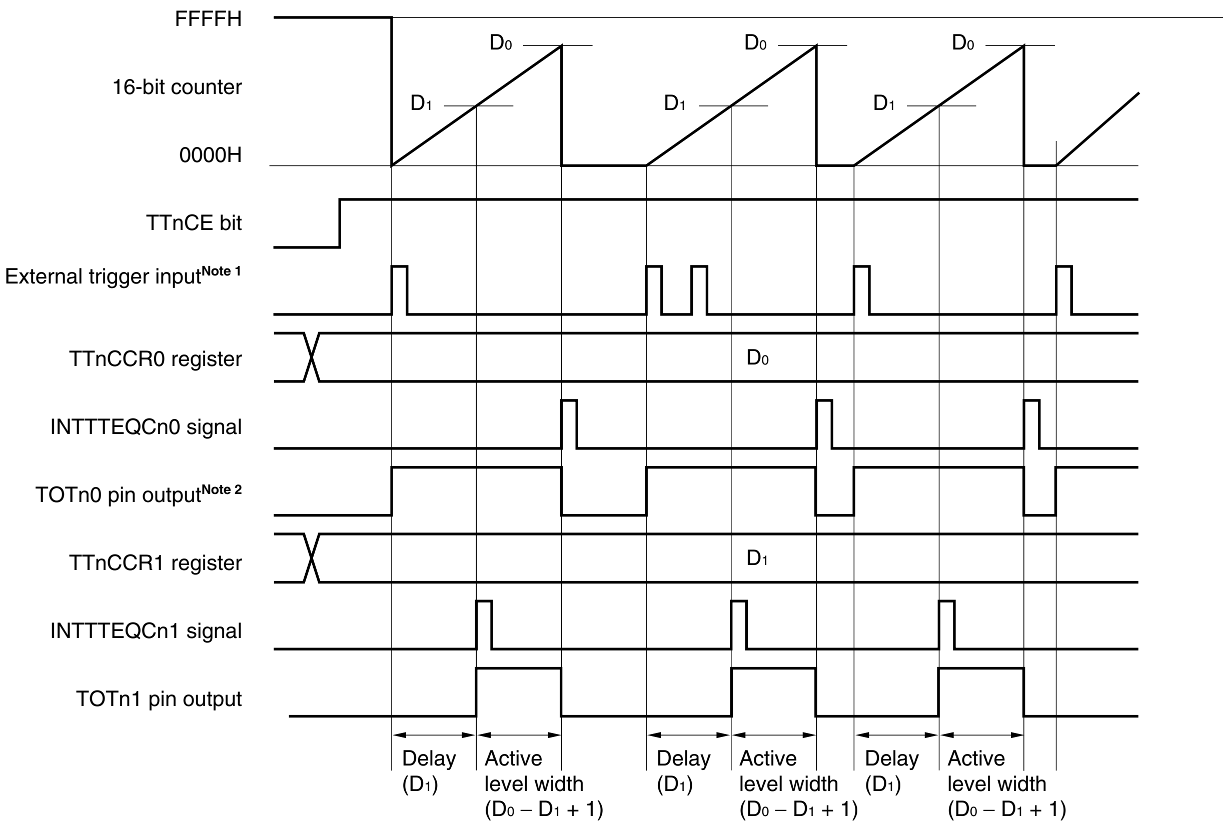
Notes 1. Because the external trigger input pin (TITk0), external event count input pin (TITk0), and timer output pin (TOTk0) share the same alternate-function pin, the two functions cannot be used at the same time.

2. Edge detector for external trigger input.
Set by the TTkIOC2.TTkETS1 and TTkIOC2.TTkETS0 bits.

3. Edge detector for external event count input.
Set by the TTkIOC2.TTkEES1 and TTkIOC2.TTkEES0 bits.

Remark k = 2, 3

Figure 8-29. Basic Timing in One-Shot Pulse Output Mode



Notes 1. EVTTm pin input in the case of TMT0 and TMT1, and TITk0 pin input in the case of TMT2 and TMT3.

2. In the case of TMT2 and TMT3, this function can only be used by using a software trigger.

When the TTnCE bit is set to 1, 16-bit timer/event counter T waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOTn1 pin. After the one-shot pulse is output, the 16-bit counter is cleared to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

Output delay period = (Set value of TTnCCR1 register) × Count clock cycle
Active level width = (Set value of TTnCCR0 register − Set value of TTnCCR1 register + 1) × Count clock cycle

The compare match interrupt request signal (INTTTEQCn0) is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal (INTTTEQCn1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The valid edge of an external trigger input (EVTTm pin in the case of TMT0 and TMT1, and TITk0 pin in the case of TMT2 and TMT3) or setting the software trigger (TTnCTL1.TTnEST bit) to 1 is used as the trigger.

Remark n = 0 to 3
m = 0, 1
k = 2, 3

Figure 8-30. Setting of Registers in One-Shot Pulse Output Mode (1/2)

(a) TMTn control register 0 (TTnCTL0)

	TTnCE					TTnCKS2	TTnCKS1	TTnCKS0
TTnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TTnCKS2 to TTnCKS0: Select count clock[Note]

TTnCE:
0: Stop counting
1: Enable counting

Note The setting is invalid when the TTnCTL1.TTnEEE bit = 1.

(b) TMTn control register 1 (TTnCTL1)

		TTnEST	TTnEEE		TTnMD3	TTnMD2	TTnMD1	TTnMD0
TTnCTL1	0	0/1	0/1	0	0	0	1	1

TTnMD3 to TTnMD0:
0, 0, 1, 1:
One-shot pulse output mode

TTnEEE:
0: Operate on count clock selected by TTnCKS0 to TTnCKS2 bits
1: Count external event input signal

TTnEST: Generate software trigger when 1 is written

(c) TMTn I/O control register 0 (TTnIOC0)

					TTnOL1	TTnOE1	TTnOL0	TTnOE0
TTnIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

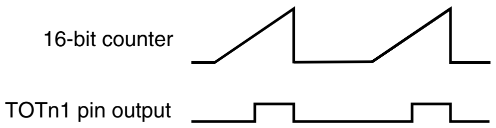
TTnOE0:
0: Disable TOTn0 pin output
1: Enable TOTn0 pin output

TTnOL0: Setting of TOTn0 pin output level while waiting for external trigger
0: Low level
1: High level

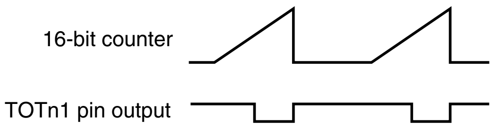
TTnOE1:
0: Disable TOTn1 pin output
1: Enable TOTn1 pin output

TTnOL1: Setting of TOTn1 pin output level while waiting for external trigger
0: Low level
1: High level

- When TTnOL1 bit = 0



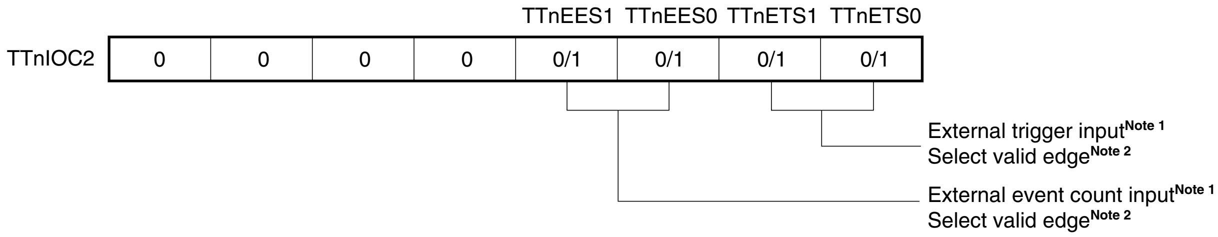
- When TTnOL1 bit = 1



Note Set this bit to 0 when not using the TOTn0 pin in the one-shot pulse output mode.

Figure 8-30. Setting of Registers in One-Shot Pulse Output Mode (2/2)

(d) TMTn I/O control register 2 (TTnIOC2)


TTnEES1
TTnEES0
TTnETS1
TTnETS0
TTnIOC2
0
0
0
0
0/1
0/1
0/1
0/1
External trigger input[Note 1]
Select valid edge[Note 2]
External event count input[Note 1]
Select valid edge[Note 2]

Notes 1. TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

2. Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(e) TMTn counter read buffer register (TTnCNT)

The value of the 16-bit counter can be read by reading the TTnCNT register.

(f) TMTn capture/compare registers 0 and 1 (TTnCCR0 and TTnCCR1)

If D_0 is set to the TTnCCR0 register and D_1 to the TTnCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width = (D_0 − D_1 + 1) × Count clock cycle

Output delay period = D_1 × Count clock cycle

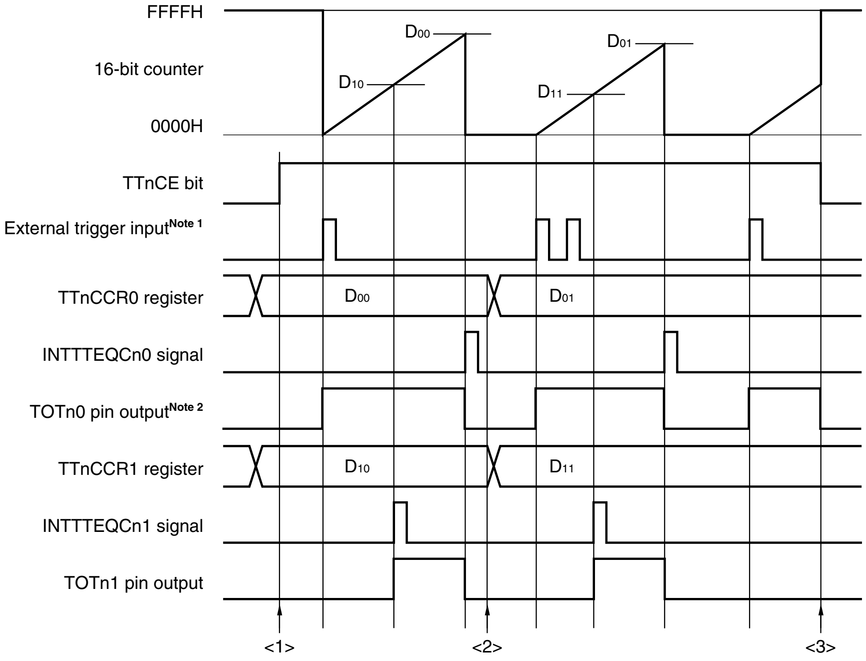
Caution One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TTnCCR1 register is greater than that set in the TTnCCR0 register.

Remarks 1. TMTm control register 2 (TTmCTL2), TMTn I/O control register 1 (TTnIOC1), TMTm I/O control register 3 (TTmIOC3), TMTn option register 0 (TTnOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the one-shot pulse output mode.

2. n = 0 to 3
m = 0, 1
k = 2, 3

(1) Operation flow in one-shot pulse output mode

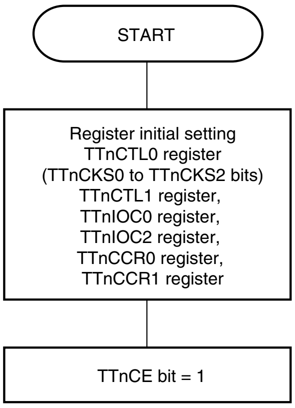
Figure 8-31. Software Processing Flow in One-Shot Pulse Output Mode



Notes 1. EVTTm pin input in the case of TMT0 and TMT1, and TITk0 pin input in the case of TMT2 and TMT3.

2. In the case of TMT2 and TMT3, this function can only be used by using a software trigger.

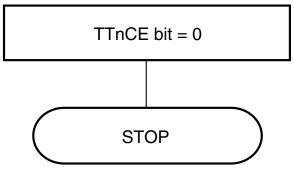
<1> Count operation start flow



Initial setting of these registers is performed before setting the TTnCE bit to 1.

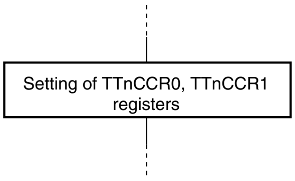
The TTnCKS0 to TTnCKS2 bits can be set at the same time as when counting starts (TTnCE bit = 1).
Trigger wait status

<3> Count operation stop flow



Count operation is stopped

<2> TTnCCR0, TTnCCR1 register setting change flow



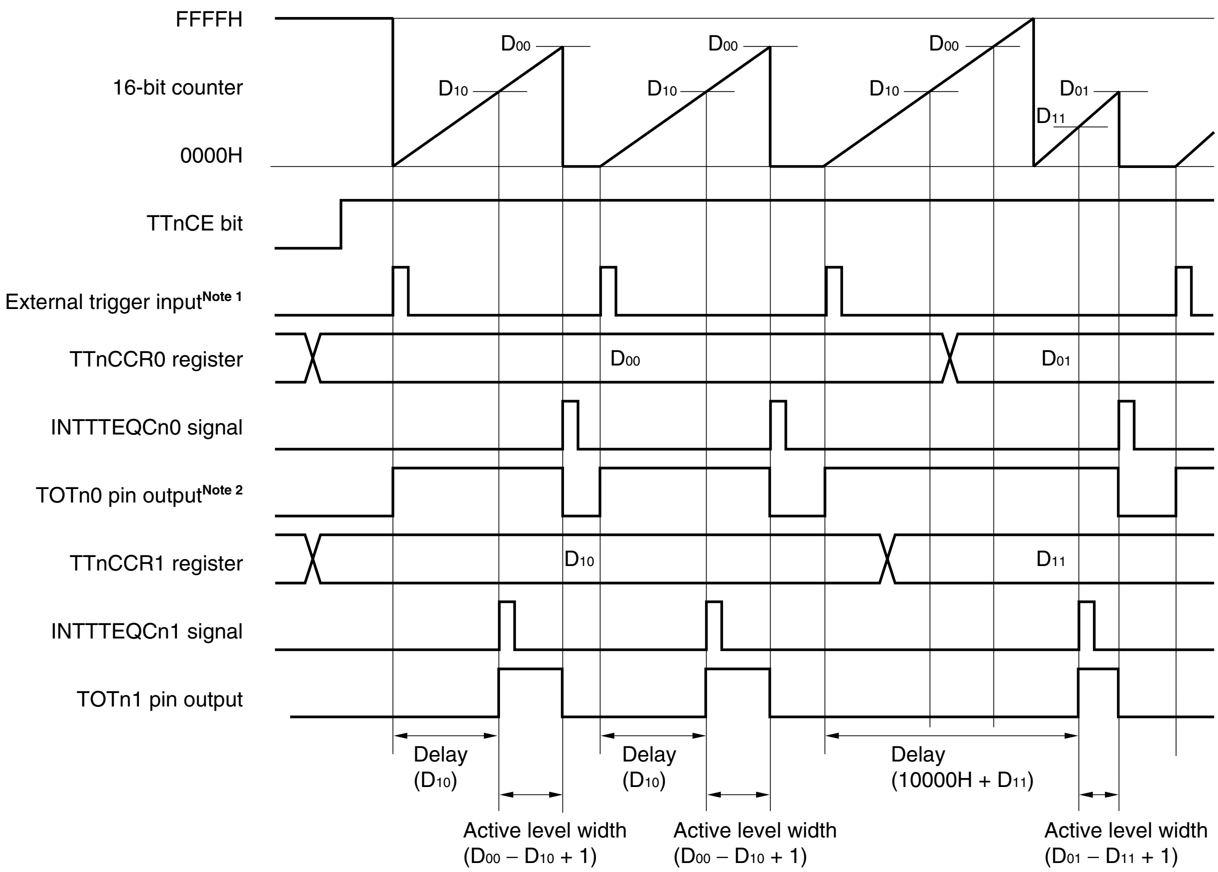
As rewriting the TTnCCRa register immediately forwards to the CCRa buffer register, rewriting immediately after the generation of the INTTTEQCn0 signal is recommended.

Remark n = 0 to 3
a = 0, 1

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TTnCCRa register

If the value of the TTnCCRa register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



Notes 1. TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

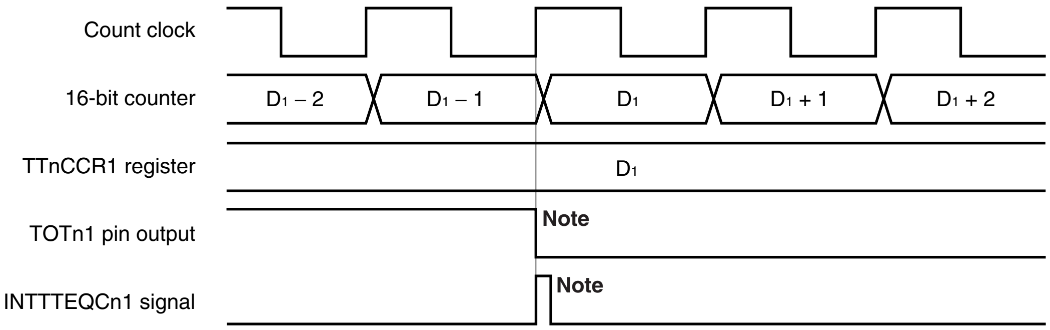
2. In the case of TMT2 and TMT3, this function can only be used by using a software trigger.

When the TTnCCR0 register is rewritten from D_{00} to D_{01} and the TTnCCR1 register from D_{10} to D_{11} where $D_{00} > D_{01}$ and $D_{10} > D_{11}$, if the TTnCCR1 register is rewritten when the count value of the 16-bit counter is greater than D_{11} and less than D_{10} and if the TTnCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{11}, the counter generates the INTTTEQCn1 signal and asserts the TOTn1 pin. When the count value matches D_{01}, the counter generates the INTTTEQCn0 signal, deasserts the TOTn1 pin, and stops counting.
Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark n = 0 to 3
a = 0, 1

(b) Generation timing of compare match interrupt request signal (INTTTEQCn1)

The generation timing of the INTTTEQCn1 signal in the one-shot pulse output mode is different from INTTTEQCn1 signals in other mode; the INTTTEQCn1 signal is generated when the count value of the 16-bit counter matches the value of the TTnCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 3

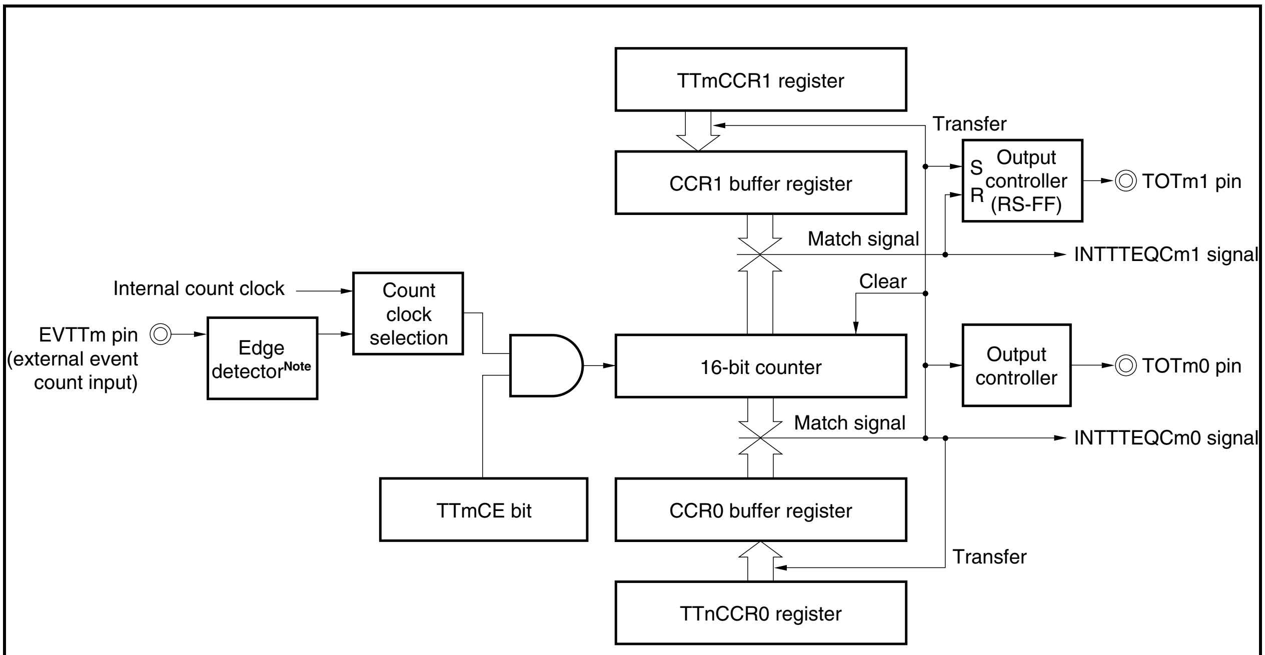
Usually, the INTTTEQCn1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TTnCCR1 register.
In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOTn1 pin.

8.6.5 PWM output mode (TTnMD3 to TTnMD0 bits = 0100)

In the PWM output mode, a PWM waveform is output from the TOTn1 pin when the TTnCTL0.TTnCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TTnCCR0 register + 1 as half its cycle is output from the TOTn0 pin.

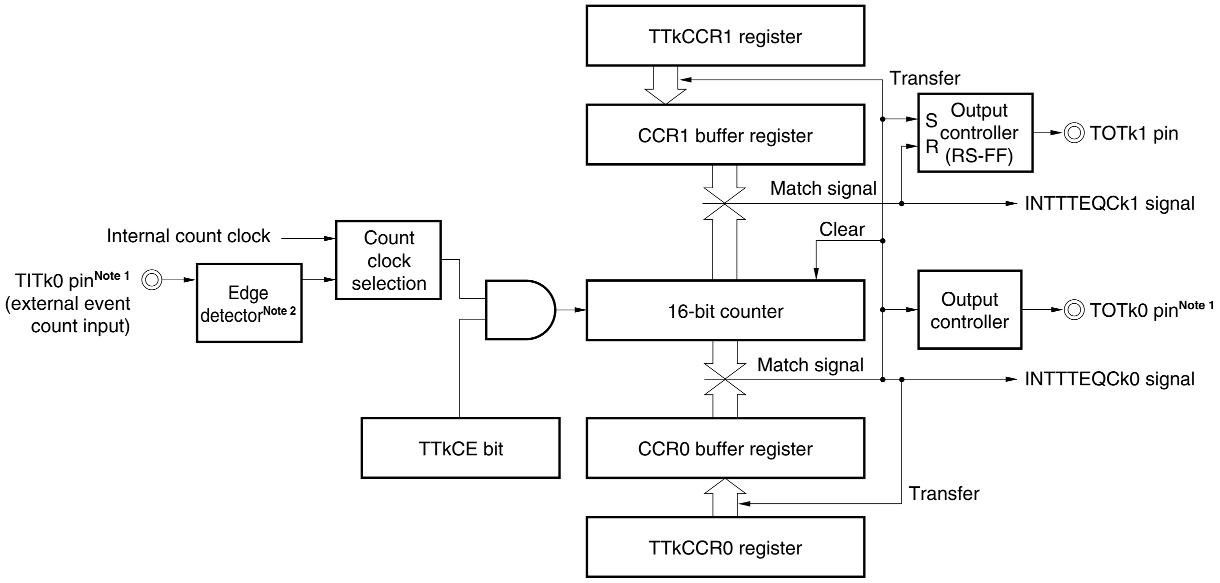
Figure 8-32. Configuration of TMT0 and TMT1 in PWM Output Mode



Note Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark m = 0, 1

Figure 8-33. Configuration of TMT2 and TMT3 in PWM Output Mode

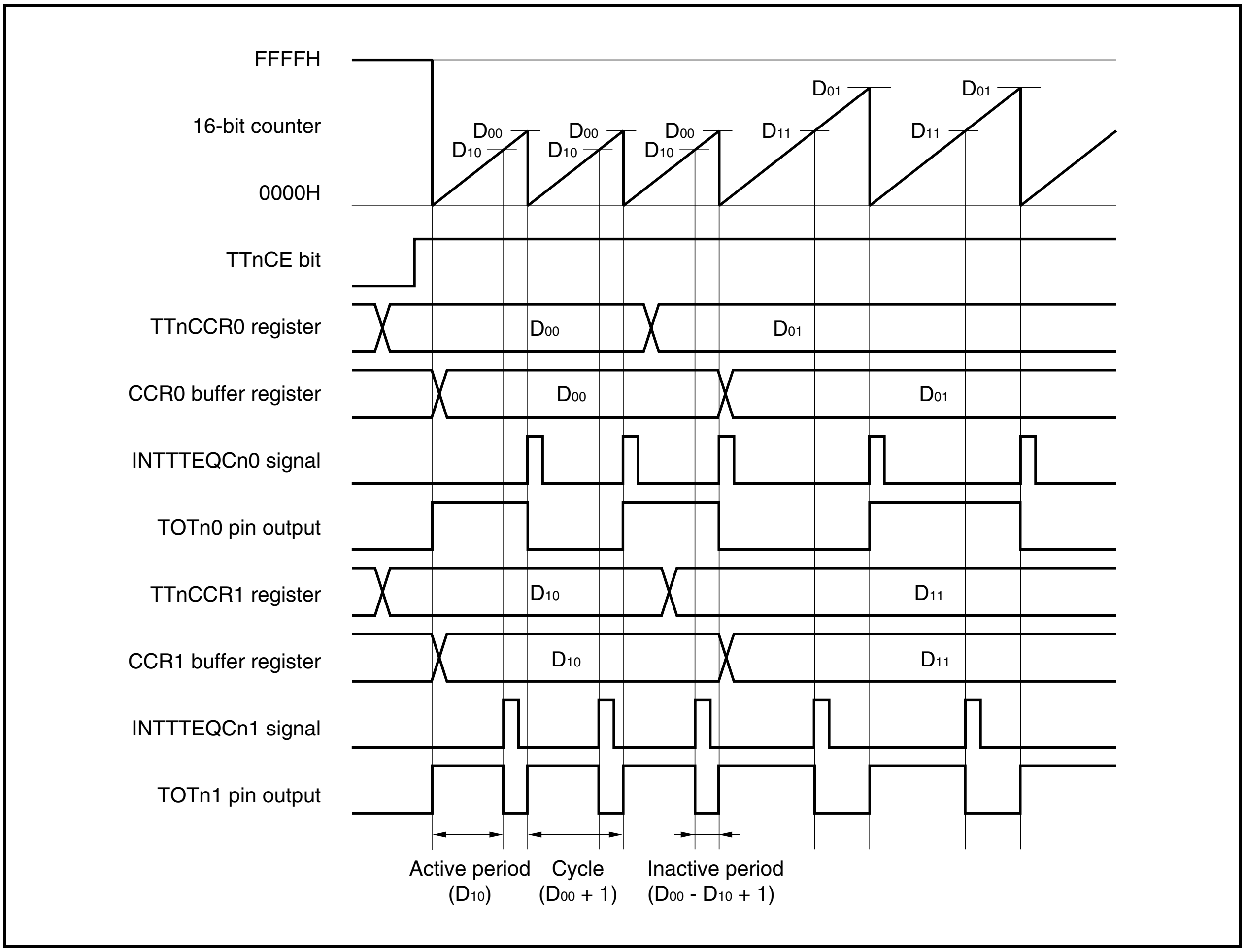


Notes 1. The external event count input pin (TITk0) is also used as the timer output pin (TOTk0), so these functions cannot be used at the same time.

2. Set by the TTkIOC2.TTkEES1 and TTkIOC2.TTkEES0 bits.

Remark k = 2, 3

Figure 8-34. Basic Timing in PWM Output Mode



When the TTnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOTn1 pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TTnCCR1 register) × Count clock cycle
Cycle = (Set value of TTnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TTnCCR1 register)/(Set value of TTnCCR0 register + 1)

The PWM waveform can be changed by rewriting the TTnCCRa register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal (INTTTEQCn0) is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal (INTTTEQCn1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TTnCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

Remark n = 0 to 3
a = 0, 1

Figure 8-35. Setting of Registers in PWM Output Mode (1/2)

(a) TMTn control register 0 (TTnCTL0)

	TTnCE					TTnCKS2	TTnCKS1	TTnCKS0
TTnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TTnCKS2 to TTnCKS0: Select count clock[Note]

TTnCE:
0: Stop counting
1: Enable counting

Note The setting is invalid when the TTnCTL1.TTnEEE bit = 1.

(b) TMTn control register 1 (TTnCTL1)

		TTnEST	TTnEEE		TTnMD3	TTnMD2	TTnMD1	TTnMD0
TTnCTL1	0	0	0/1	0	0	1	0	0

TTnMD3 to TTnMD0:
0, 1, 0, 0:
PWM output mode

TTnEEE:
0: Operate on count clock selected by TTnCKS0 to TTnCKS2 bits
1: Count with external event count input signal

(c) TMTn I/O control register 0 (TTnIOC0)

					TTnOL1	TTnOE1	TTnOL0	TTnOE0
TTnIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

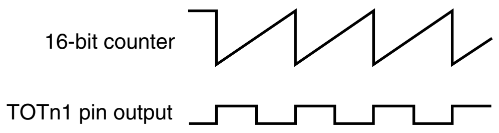
TTnOE0:
0: Disable TOTn0 pin output
1: Enable TOTn0 pin output

TTnOL0:
Setting of TOTn0 pin output level before count operation
0: Low level
1: High level

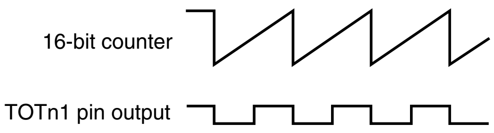
TTnOE1:
0: Disable TOTn1 pin output
1: Enable TOTn1 pin output

TTnOL1:
Setting of TOTn1 pin output level before count operation
0: Low level
1: High level

- When TTnOL1 bit = 0



- When TTnOL1 bit = 1



Note Set this bit to 0 when not using the TOTn0 pin in the PWM output mode.

Figure 8-35. Register Setting in PWM Output Mode (2/2)

(d) TMTn I/O control register 2 (TTnIOC2)

					TTnEES1	TTnEES0	TTnETS1	TTnETS0
TTnIOC2	0	0	0	0	0/1	0/1	0	0

TTnEES1, TTnEES0: Select valid edge of external event count input[Note]

Note TMT0 and TMT1: EVTTm pin input
TMT2 and TMT3: TITk0 pin input

(e) TMTn counter read buffer register (TTnCNT)

The value of the 16-bit counter can be read by reading the TTnCNT register.

(f) TMTn capture/compare registers 0 and 1 (TTnCCR0 and TTnCCR1)

If D_0 is set to the TTnCCR0 register and D_1 to the TTnCCR1 register, the cycle and active level of the PWM waveform are as follows.

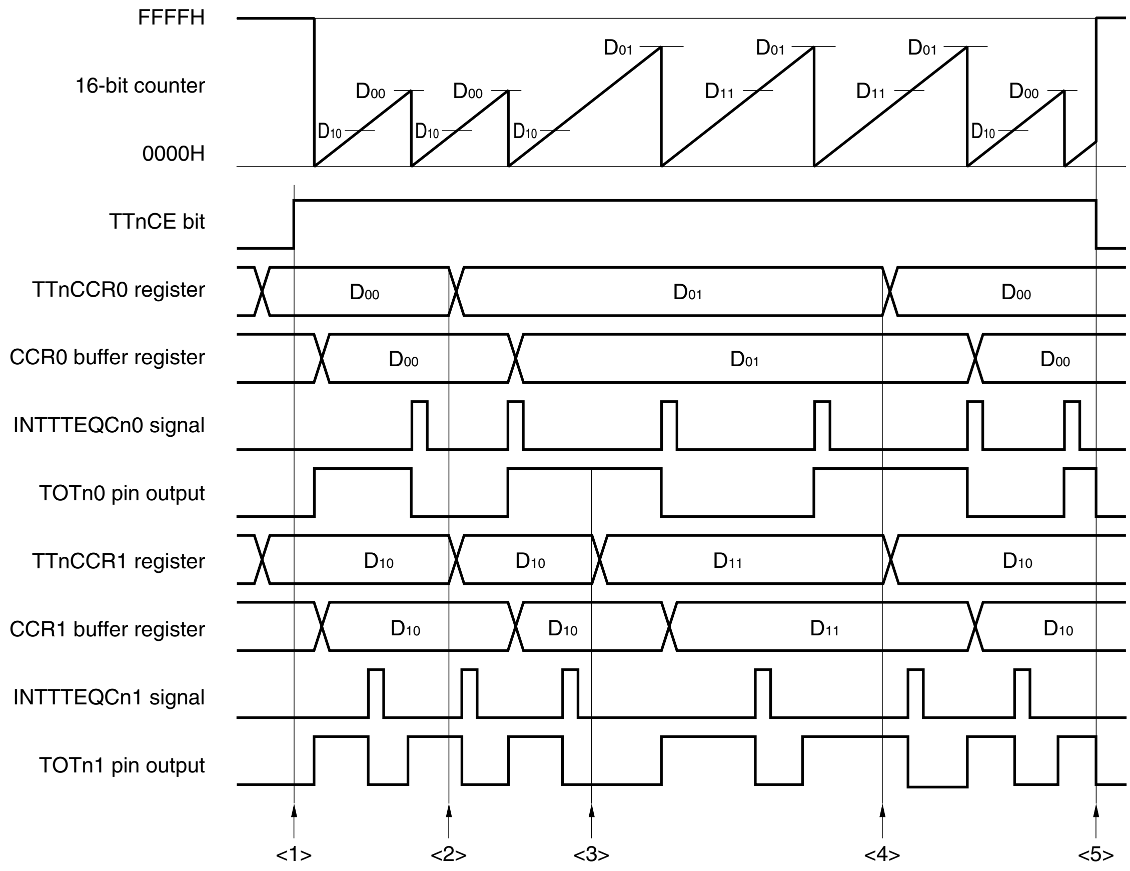
Cycle = (D_0 + 1) × Count clock cycle
Active level width = D_1 × Count clock cycle

Remarks **1.** TMTm control register 2 (TTmCTL2), TMTn I/O control register 1 (TTnIOC1), TMTm I/O control register 3 (TTnCTL3), TMTn option register 0 (TTnOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the PWM output mode.

2. n = 0 to 3
m = 0, 1
k = 2, 3

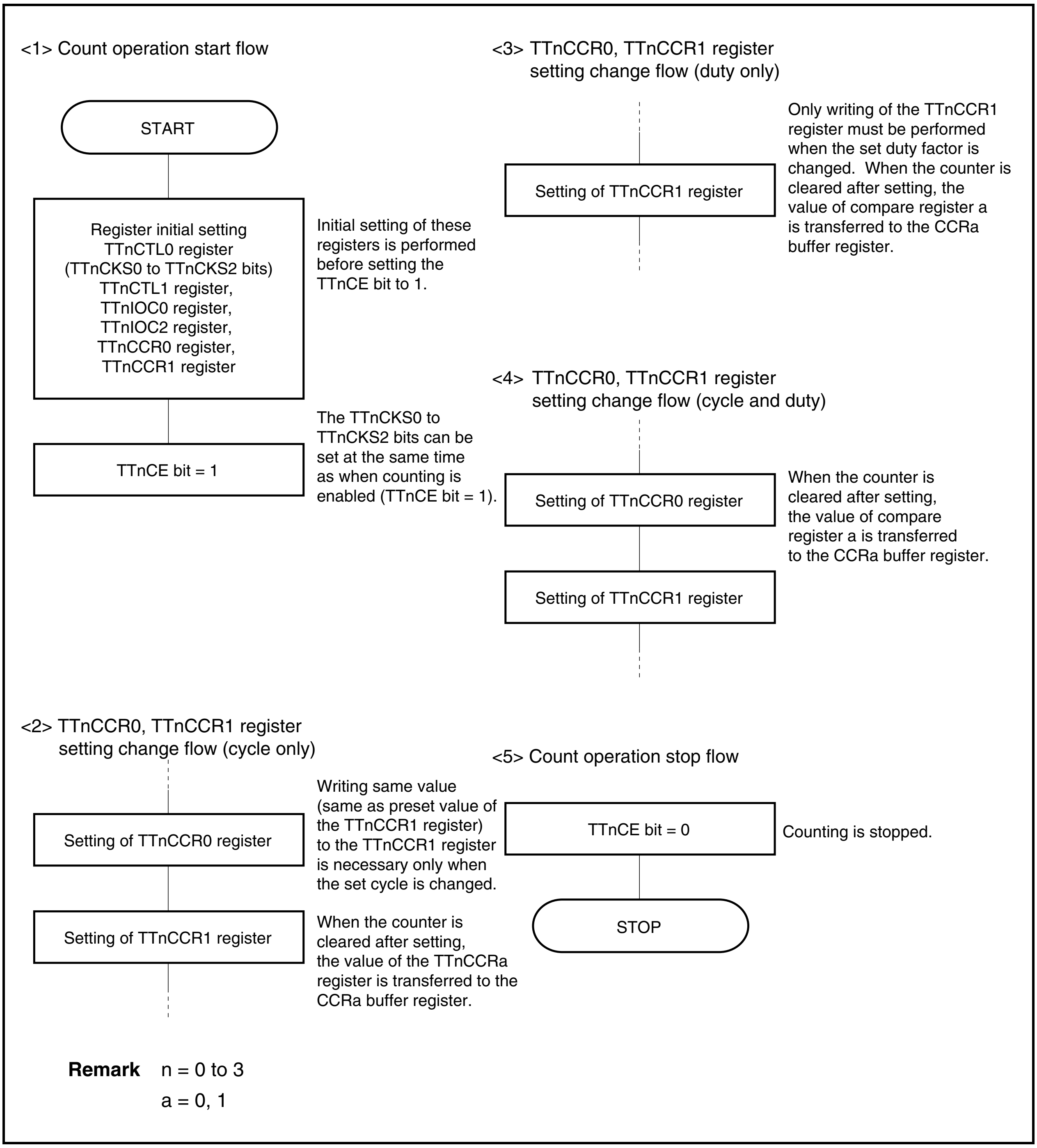
(1) Operation flow in PWM output mode

Figure 8-36. Software Processing Flow in PWM Output Mode (1/2)



Remark n = 0 to 3

Figure 8-36. Software Processing Flow in PWM Output Mode (2/2)


START
Register initial setting
TTnCTL0 register
(TTnCKS0 to TTnCKS2 bits)
TTnCTL1 register,
TTnIOC0 register,
TTnIOC2 register,
TTnCCR0 register,
TTnCCR1 register
TTnCE bit = 1
Setting of TTnCCR0 register
Setting of TTnCCR1 register
Setting of TTnCCR1 register
Setting of TTnCCR0 register
Setting of TTnCCR1 register
TTnCE bit = 0
STOP

<1> Count operation start flow

Initial setting of these registers is performed before setting the TTnCE bit to 1.

The TTnCKS0 to TTnCKS2 bits can be set at the same time as when counting is enabled (TTnCE bit = 1).

<2> TTnCCR0, TTnCCR1 register setting change flow (cycle only)

Writing same value (same as preset value of the TTnCCR1 register) to the TTnCCR1 register is necessary only when the set cycle is changed.

When the counter is cleared after setting, the value of the TTnCCRa register is transferred to the CCRa buffer register.

<3> TTnCCR0, TTnCCR1 register setting change flow (duty only)

Only writing of the TTnCCR1 register must be performed when the set duty factor is changed. When the counter is cleared after setting, the value of compare register a is transferred to the CCRa buffer register.

<4> TTnCCR0, TTnCCR1 register setting change flow (cycle and duty)

When the counter is cleared after setting, the value of compare register a is transferred to the CCRa buffer register.

<5> Count operation stop flow

Counting is stopped.

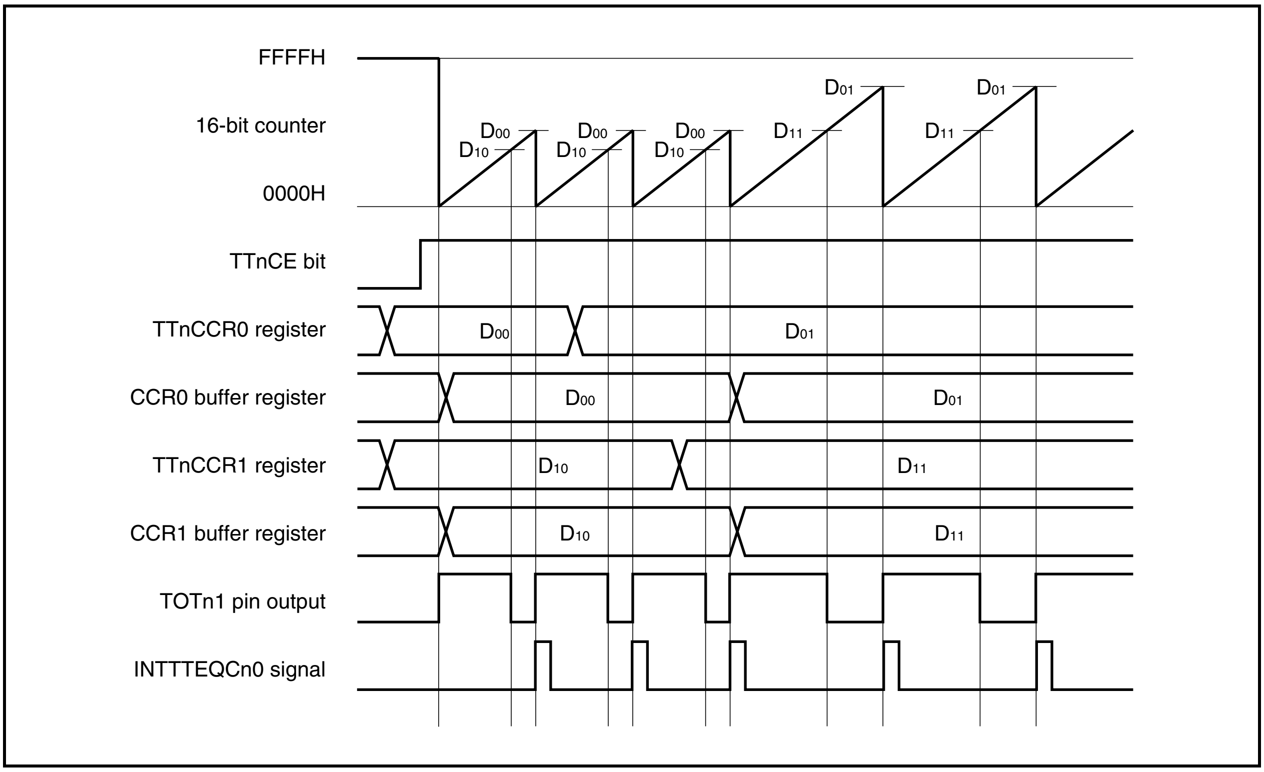
Remark n = 0 to 3
a = 0, 1

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TTnCCR1 register last.

Rewrite the TTnCCRa register after writing the TTnCCR1 register after the INTTTEQCn0 signal is detected.



To transfer data from the TTnCCRa register to the CCRa buffer register, the TTnCCR1 register must be written.

To change both the cycle and active level of the PWM waveform at this time, first set the cycle to the TTnCCR0 register and then set the active level to the TTnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TTnCCR0 register, and then write the same value (same as preset value of the TTnCCR1 register) to the TTnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TTnCCR1 register has to be set.

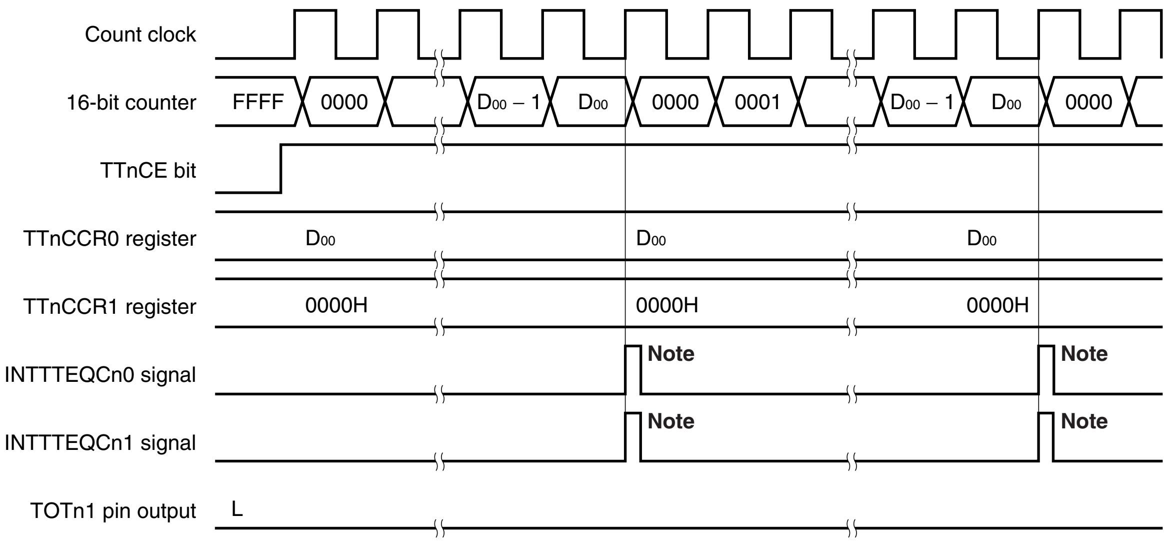
After data is written to the TTnCCR1 register, the value written to the TTnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TTnCCR0 or TTnCCR1 register again after writing the TTnCCR1 register once, do so after the INTTTEQCn0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TTnCCRa register to the CCRa buffer register conflicts with writing the TTnCCRa register.

Remark n = 0 to 3
a = 0, 1

(b) 0%/100% output of PWM waveform

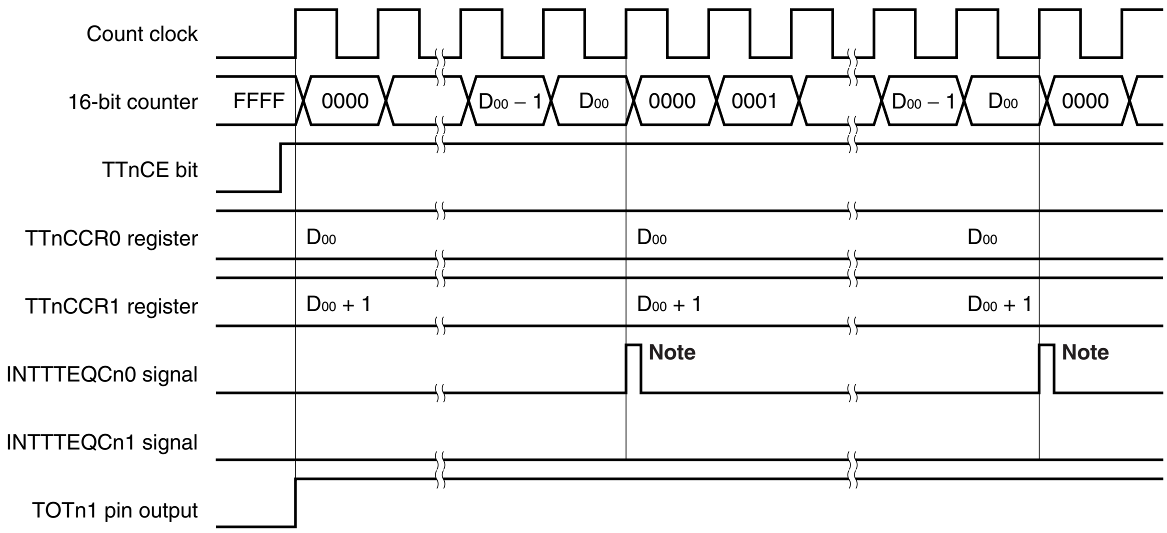
To output a 0% waveform, set the TTnCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTTEQCn0 and INTTTEQCn1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 3

To output a 100% waveform, set a value of (set value of TTnCCR0 register + 1) to the TTnCCR1 register. If the set value of the TTnCCR0 register is FFFFH, 100% output cannot be produced.

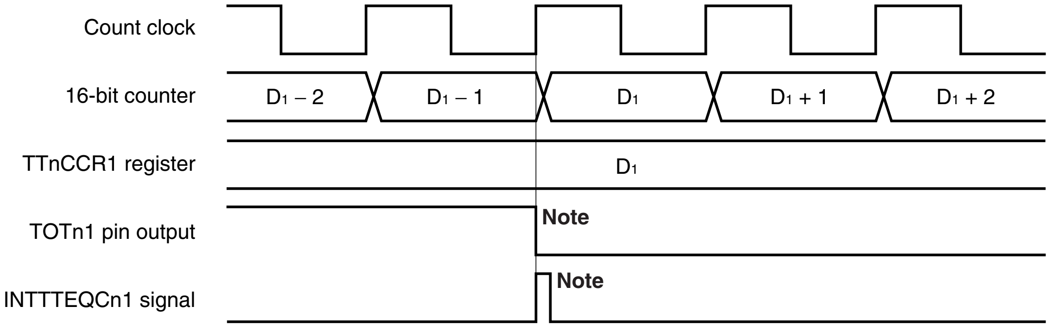


Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 3

(c) Generation timing of compare match interrupt request signal (INTTTEQCn1)

The timing of generation of the INTTTEQCn1 signal in the PWM output mode differs from the timing of INTTTEQCn1 signals in other modes; the INTTTEQCn1 signal is generated when the count value of the 16-bit counter matches the value of the TTnCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0 to 3

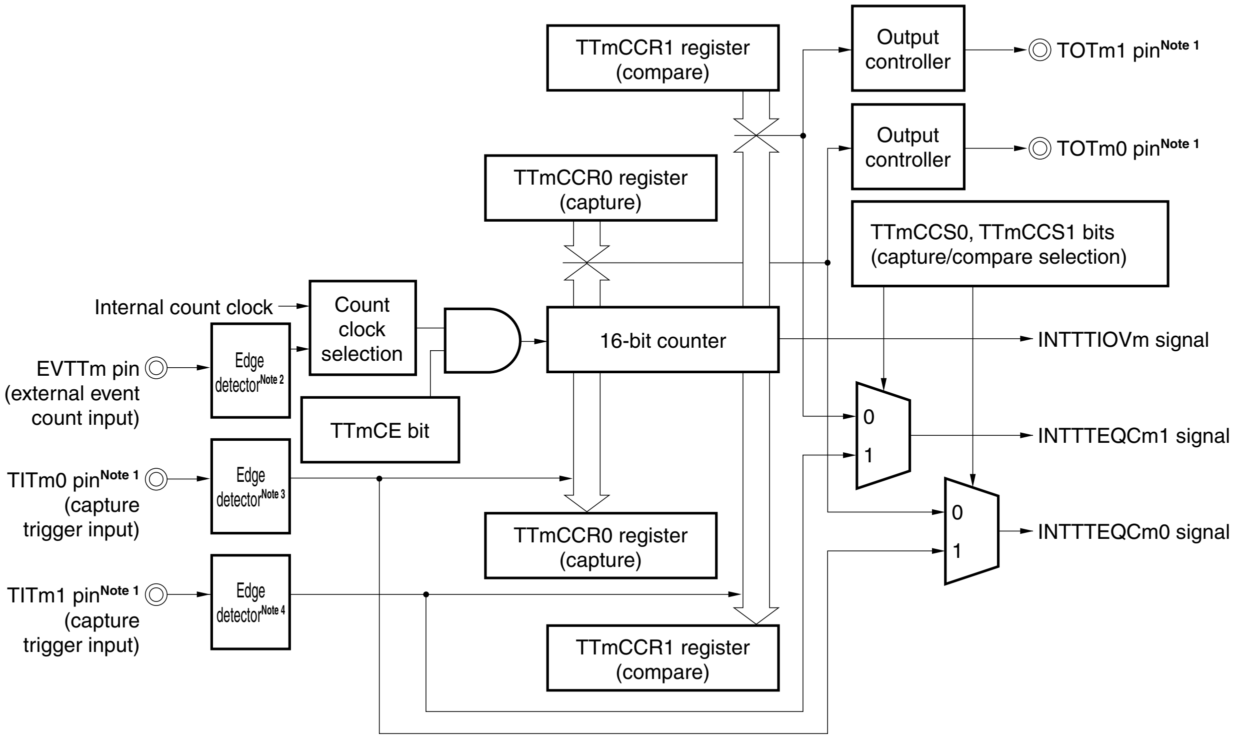
Usually, the INTTTEQCn1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TTnCCR1 register.

In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOTn1 pin.

8.6.6 Free-running timer mode (TTnMD3 to TTnMD0 bits = 0101)

In the free-running timer mode, 16-bit timer/event counter T starts counting when the TTnCTL0.TTnCE bit is set to 1. At this time, the TTnCCR0 and TTnCCR1 registers can be used as compare registers or capture registers, depending on the setting of the TTnOPT0.TTnCCS0 and TTnOPT0.TTnCCS1 bits.

Figure 8-37. Configuration of TMT0 and TMT1 in Free-Running Timer Mode



TTmCCR1 register (compare)
Output controller
TOTm1 pinNote 1
Output controller
TOTm0 pinNote 1
TTmCCR0 register (capture)
TTmCCS0, TTmCCS1 bits (capture/compare selection)
Internal count clock
Count clock selection
16-bit counter
INTTTIOVm signal
EVTTm pin (external event count input)
Edge detectorNote 2
TTmCE bit
0
1
INTTTEQCm1 signal
TITm0 pinNote 1 (capture trigger input)
Edge detectorNote 3
0
1
INTTTEQCm0 signal
TTmCCR0 register (capture)
TITm1 pinNote 1 (capture trigger input)
Edge detectorNote 4
TTmCCR1 register (compare)

Notes 1. Because the capture trigger input pins (TITm0, TITm1) and timer output pins (TOTm0, TOTm1) share the same alternate-function pins, the two functions cannot be used at the same time.

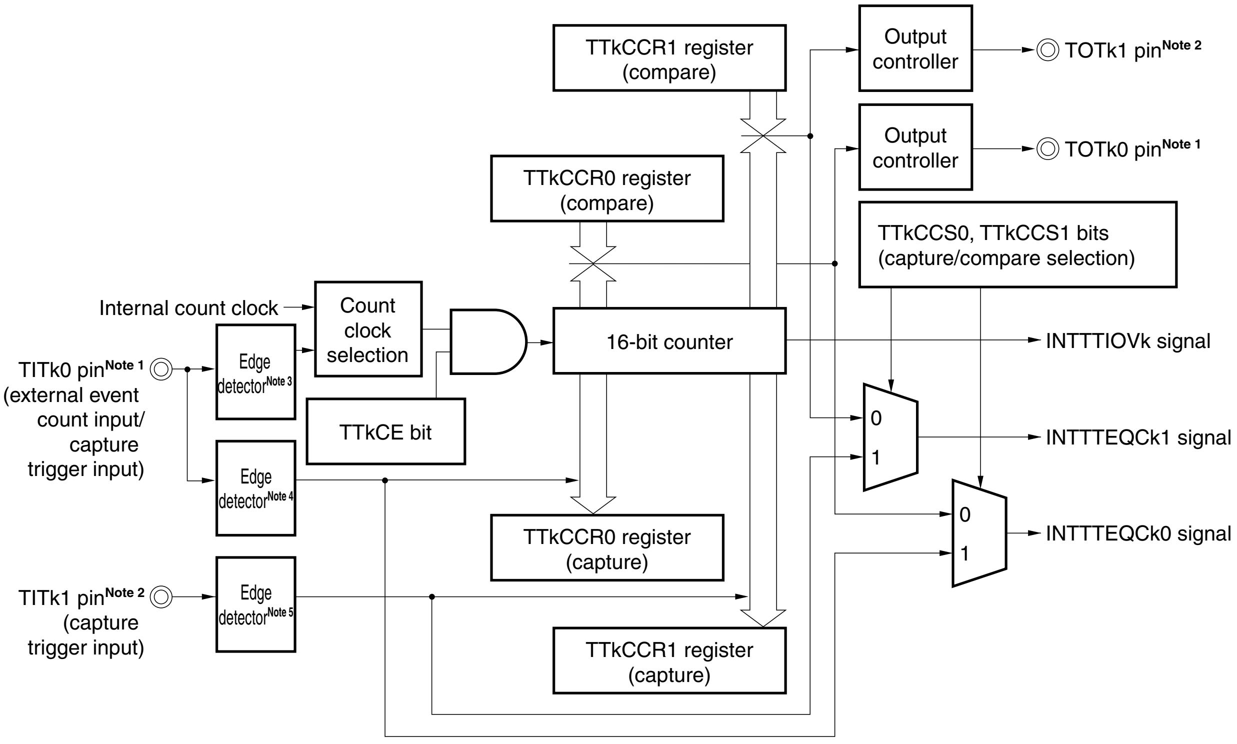
2. Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

3. Set by the TTmIOC1.TTmIS1 and TTmIOC1.TTmIS0 bits.

4. Set by the TTmIOC1.TTmIS3 and TTmIOC1.TTmIS2 bits.

Remark m = 0, 1

Figure 8-38. Configuration of TMT2 and TMT3 in Free-Running Timer Mode


TTkCCR1 register (compare)
Output controller
TOTk1 pinNote 2
Output controller
TOTk0 pinNote 1
TTkCCR0 register (compare)
TTkCCS0, TTkCCS1 bits (capture/compare selection)
Internal count clock
Count clock selection
16-bit counter
INTTTIOVk signal
TITk0 pinNote 1 (external event count input/ capture trigger input)
Edge detectorNote 3
TTkCE bit
0
1
INTTTEQCk1 signal
Edge detectorNote 4
0
1
INTTTEQCk0 signal
TTkCCR0 register (capture)
TITk1 pinNote 2 (capture trigger input)
Edge detectorNote 5
TTkCCR1 register (capture)

Notes 1. Because the external event count input pin (TITk0), capture trigger input pin (TITk0), and timer output pin (TOTk0) share the same pins, these functions cannot be used at the same time.

2. Because the capture trigger input pin (TITk1) and timer output pin (TOTk1) share the same pins, the two functions cannot be used at the same time.

3. Set by the TTkIOC2.TTkEES1 and TTkIOC2.TTkEES0 bits.

4. Set by the TTkIOC1.TTkIS1 and TTkIOC1.TTkIS0 bits.

5. Set by the TTkIOC1.TTkIS3 and TTkIOC1.TTkIS2 bits.

Remark k = 2, 3

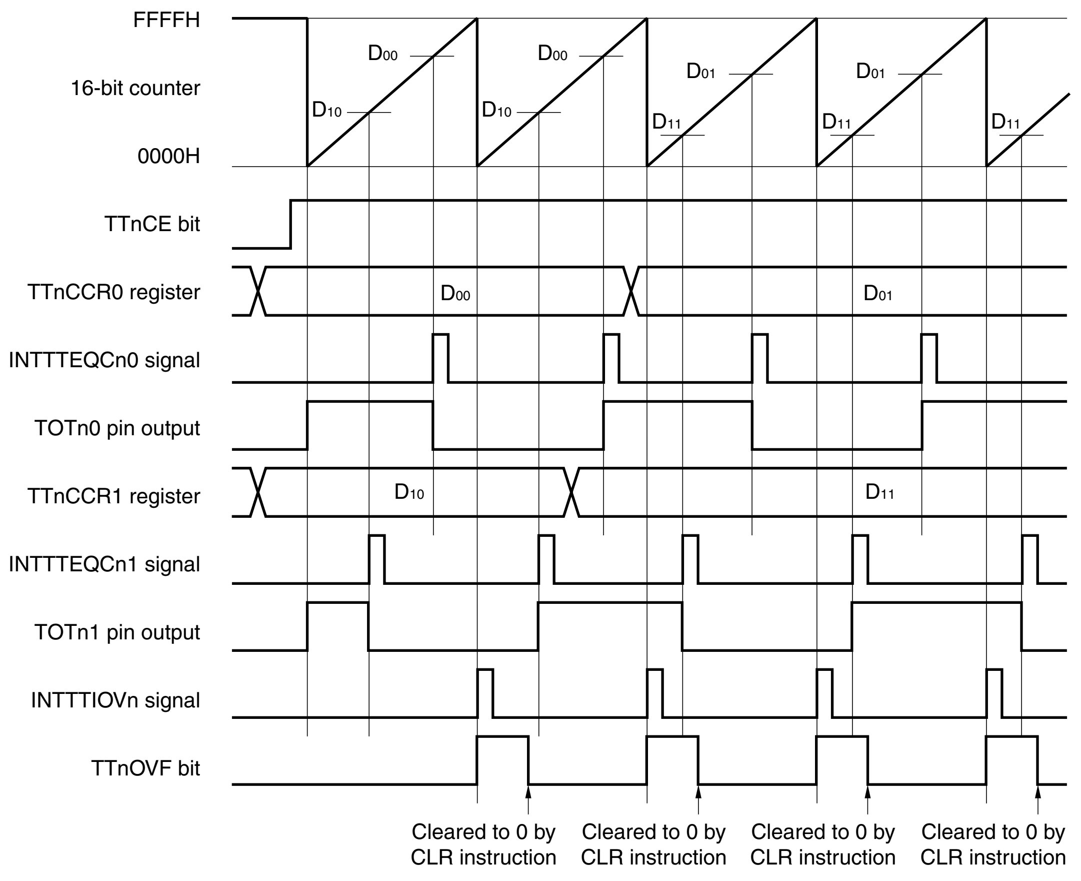
- Compare operation

  When the TTnCE bit is set to 1, 16-bit timer/event counter T starts counting, and the output signal of the TOTna pin is inverted. When the count value of the 16-bit counter later matches the set value of the TTnCCRa register, a compare match interrupt request signal (INTTTEQCna) is generated, and the output signal of the TOTna pin is inverted.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTTIOVn) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TTnOPT0.TTnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

  The TTnCCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time by anytime write, and compared with the count value.

Figure 8-39. Basic Timing in Free-Running Timer Mode (Compare Function)



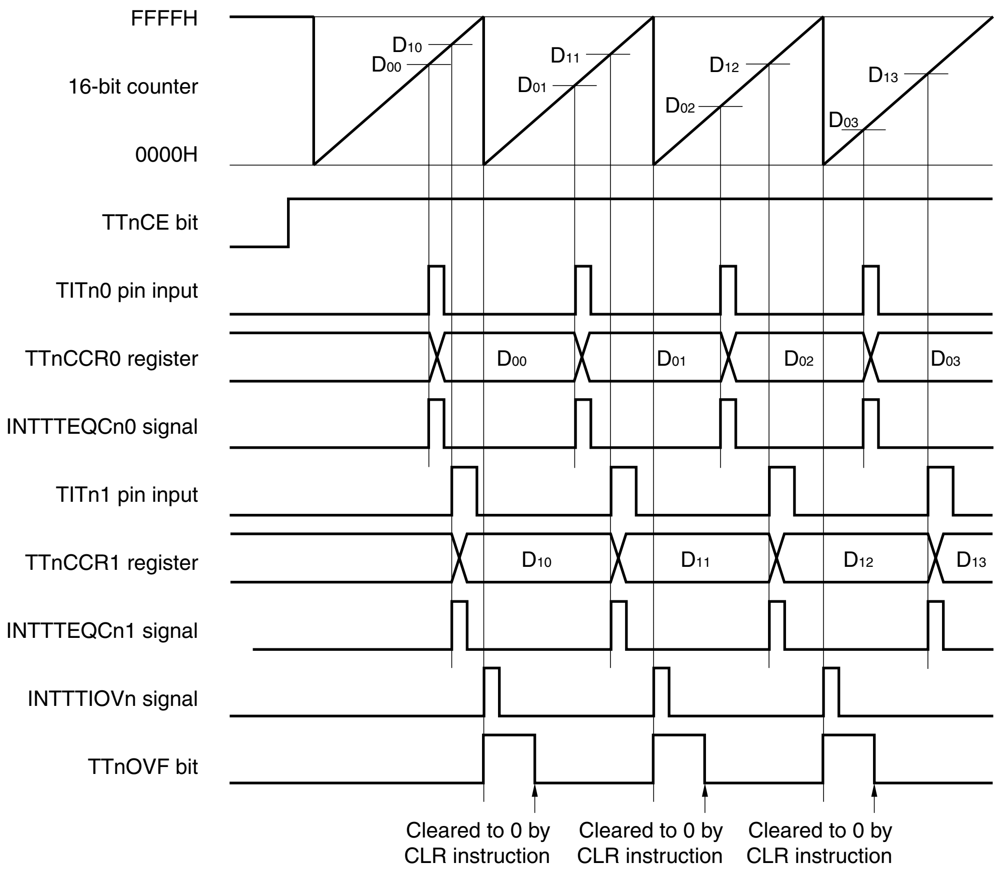
Remark n = 0 to 3
a = 0, 1

- Capture operation

  When the TTnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TITna pin is detected, the count value of the 16-bit counter is stored in the TTnCCRa register, and a capture interrupt request signal (INTTTEQCna) is generated.

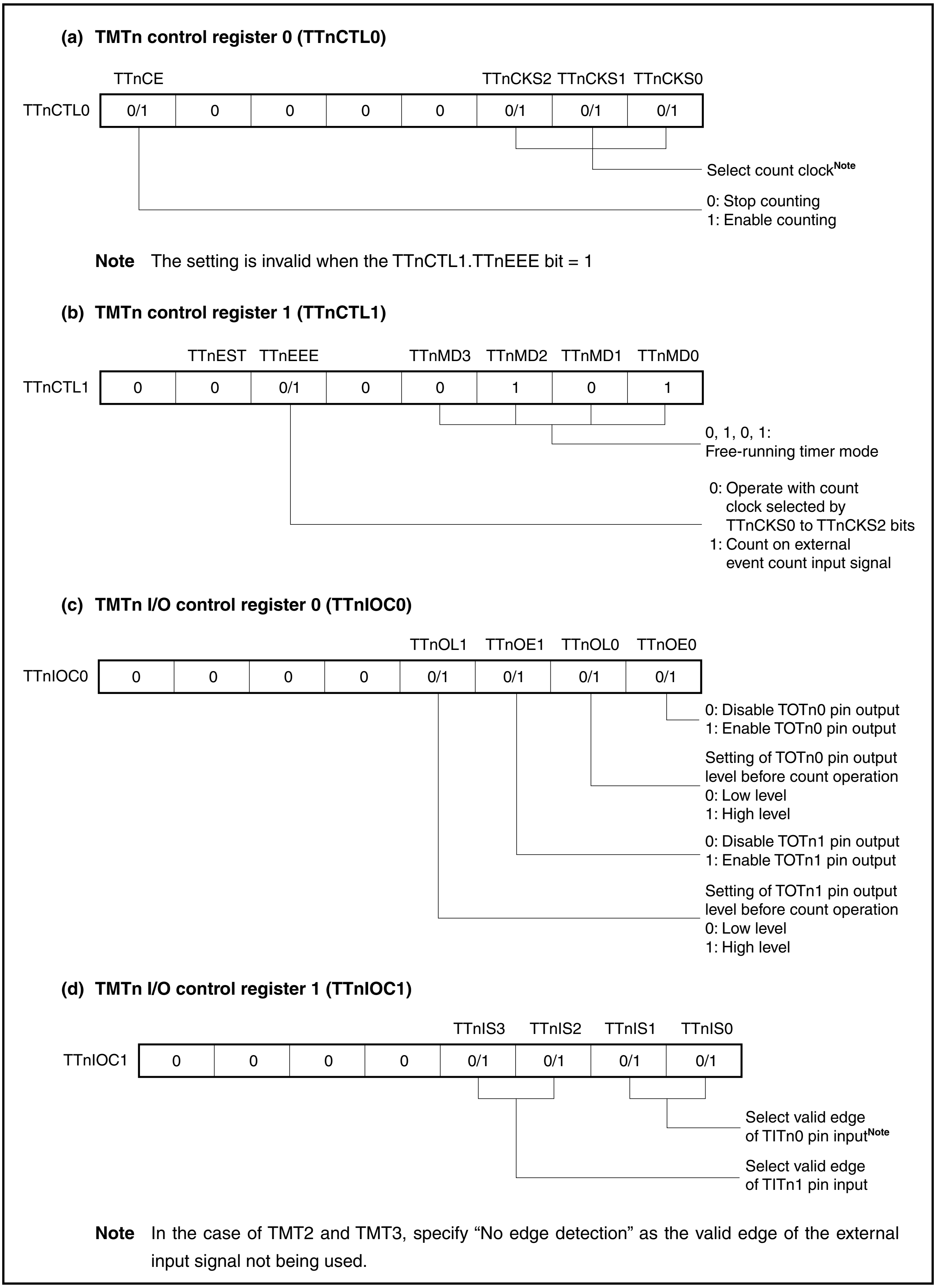
  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTTIOVn) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TTnOPT0.TTnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

Figure 8-40. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark n = 0 to 3

Figure 8-41. Register Setting in Free-Running Timer Mode (1/2)



(a) TMTn control register 0 (TTnCTL0)

	TTnCE					TTnCKS2	TTnCKS1	TTnCKS0
TTnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock[Note]
0: Stop counting
1: Enable counting

Note The setting is invalid when the TTnCTL1.TTnEEE bit = 1

(b) TMTn control register 1 (TTnCTL1)

		TTnEST	TTnEEE		TTnMD3	TTnMD2	TTnMD1	TTnMD0
TTnCTL1	0	0	0/1	0	0	1	0	1

0, 1, 0, 1:
Free-running timer mode
0: Operate with count clock selected by TTnCKS0 to TTnCKS2 bits
1: Count on external event count input signal

(c) TMTn I/O control register 0 (TTnIOC0)

					TTnOL1	TTnOE1	TTnOL0	TTnOE0
TTnIOC0	0	0	0	0	0/1	0/1	0/1	0/1

0: Disable TOTn0 pin output
1: Enable TOTn0 pin output
Setting of TOTn0 pin output level before count operation
0: Low level
1: High level
0: Disable TOTn1 pin output
1: Enable TOTn1 pin output
Setting of TOTn1 pin output level before count operation
0: Low level
1: High level

(d) TMTn I/O control register 1 (TTnIOC1)

					TTnIS3	TTnIS2	TTnIS1	TTnIS0
TTnIOC1	0	0	0	0	0/1	0/1	0/1	0/1

Select valid edge of TITn0 pin input[Note]
Select valid edge of TITn1 pin input

Note In the case of TMT2 and TMT3, specify “No edge detection” as the valid edge of the external input signal not being used.

Figure 8-41. Register Setting in Free-Running Timer Mode (2/2)

(e) TMTn I/O control register 2 (TTnIOC2)

					TTnEES1	TTnEES0	TTnETS1	TTnETS0
TTnIOC2	0	0	0	0	0/1	0/1	0	0

TTnEES1, TTnEES0: External event count input[Note 1] Select of valid edge[Note 2]

Notes 1. TMT0 and TMT1: EVTTm pin
TMT2 and TMT3: TITk0 pin

2. In the case of TMT2 and TMT3, specify “No edge detection” as the valid edge of the external input signal not being used.

(f) TMTn option register 0 (TTnOPT0)

			TTnCCS1	TTnCCS0				TTnOVF
TTnOPT0	0	0	0/1	0/1	0	0	0	0/1

TTnOVF: Overflow flag

TTnCCS0: Specifies if TTnCCR0 register functions as capture or compare register
0: Compare register
1: Capture register

TTnCCS1: Specifies if TTnCCR1 register functions as capture or compare register
0: Compare register
1: Capture register

(g) TMTn counter read buffer register (TTnCNT)

The value of the 16-bit counter can be read by reading the TTnCNT register.

(h) TMTn capture/compare registers 0 and 1 (TTnCCR0 and TTnCCR1)

These registers function as capture registers or compare registers depending on the setting of the TTnOPT0.TTnCCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TITna pin is detected.

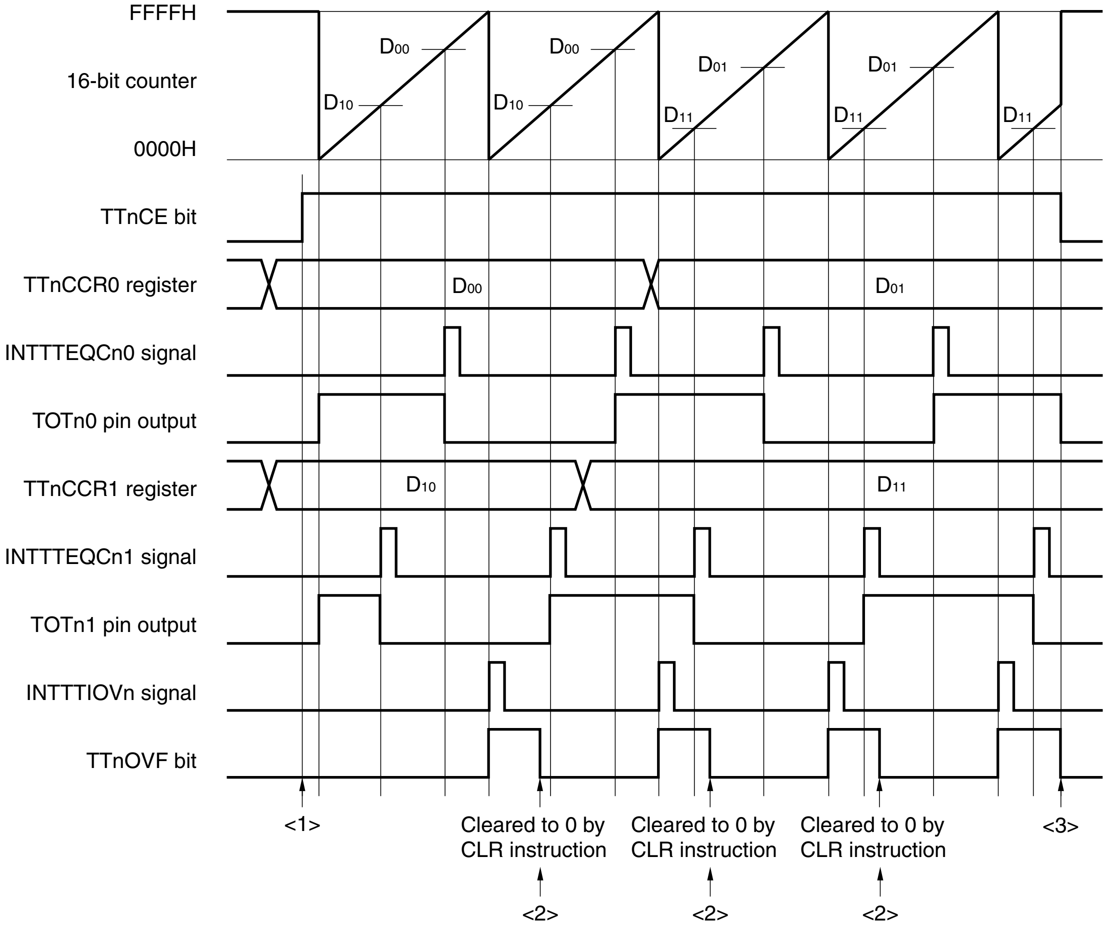
When the registers function as compare registers and when D_a is set to the TTnCCRa register, the INTTTEQCna signal is generated when the counter reaches (D_a + 1), and the output signals of the TOTn0 and TOTn1 pins are inverted.

Remark n = 0 to 3
m = 0, 1
k = 2, 3
a = 0, 1

(1) Operation flow in free-running timer mode

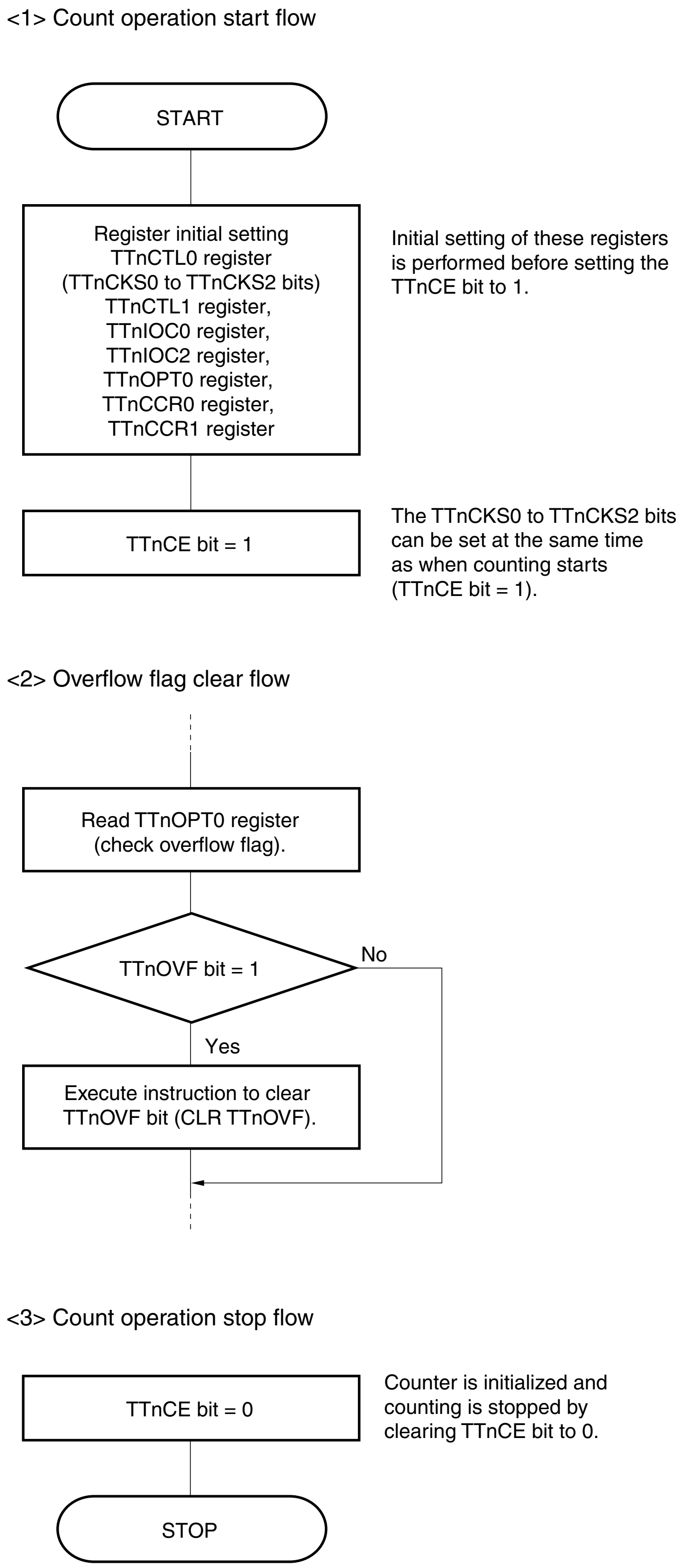
(a) When using capture/compare register as compare register

Figure 8-42. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



Remark n = 0 to 3

Figure 8-42. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



<1> Count operation start flow

Initial setting of these registers is performed before setting the TTnCE bit to 1.

The TTnCKS0 to TTnCKS2 bits can be set at the same time as when counting starts (TTnCE bit = 1).

<2> Overflow flag clear flow

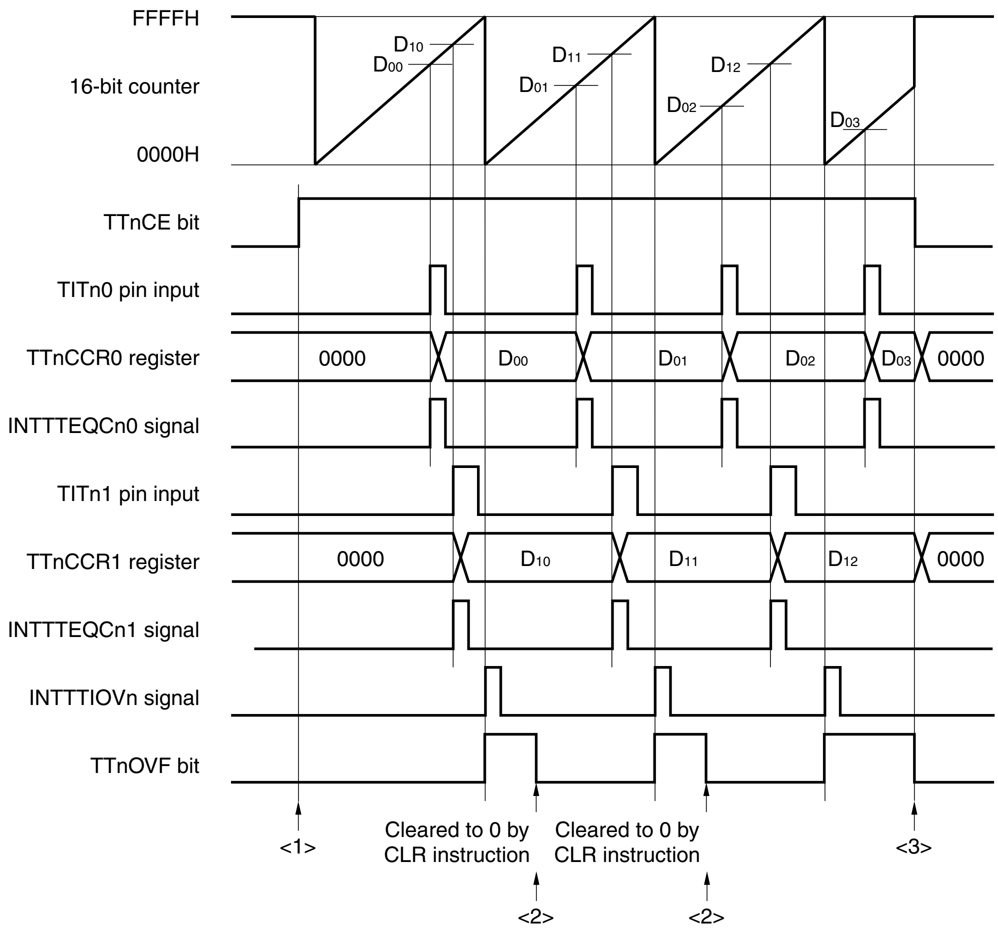
<3> Count operation stop flow

Counter is initialized and counting is stopped by clearing TTnCE bit to 0.

Remark n = 0 to 3

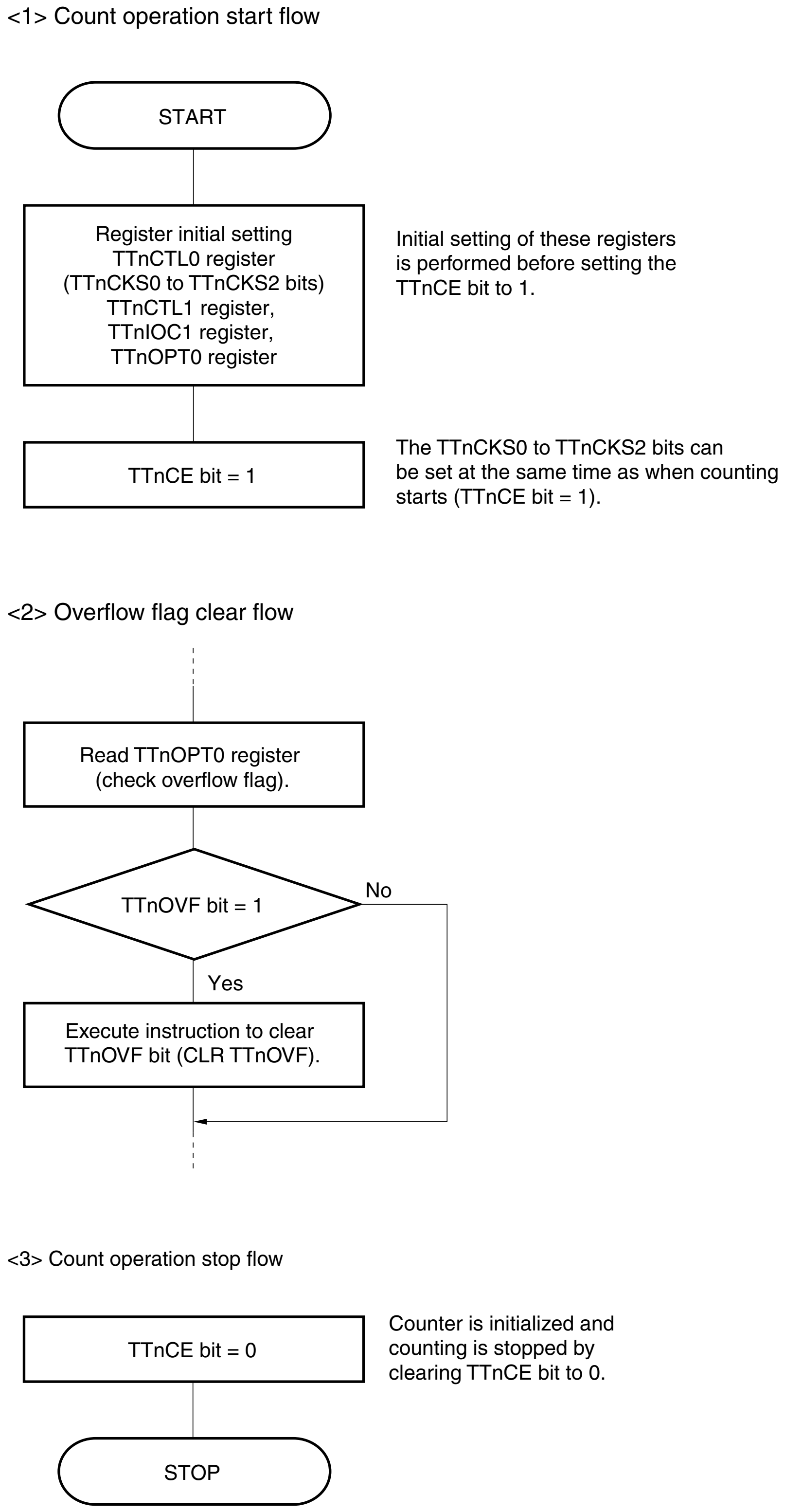
(b) When using capture/compare register as capture register

Figure 8-43. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)



Remark n = 0 to 3

Figure 8-43. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



Initial setting of these registers is performed before setting the TTnCE bit to 1.

The TTnCKS0 to TTnCKS2 bits can be set at the same time as when counting starts (TTnCE bit = 1).

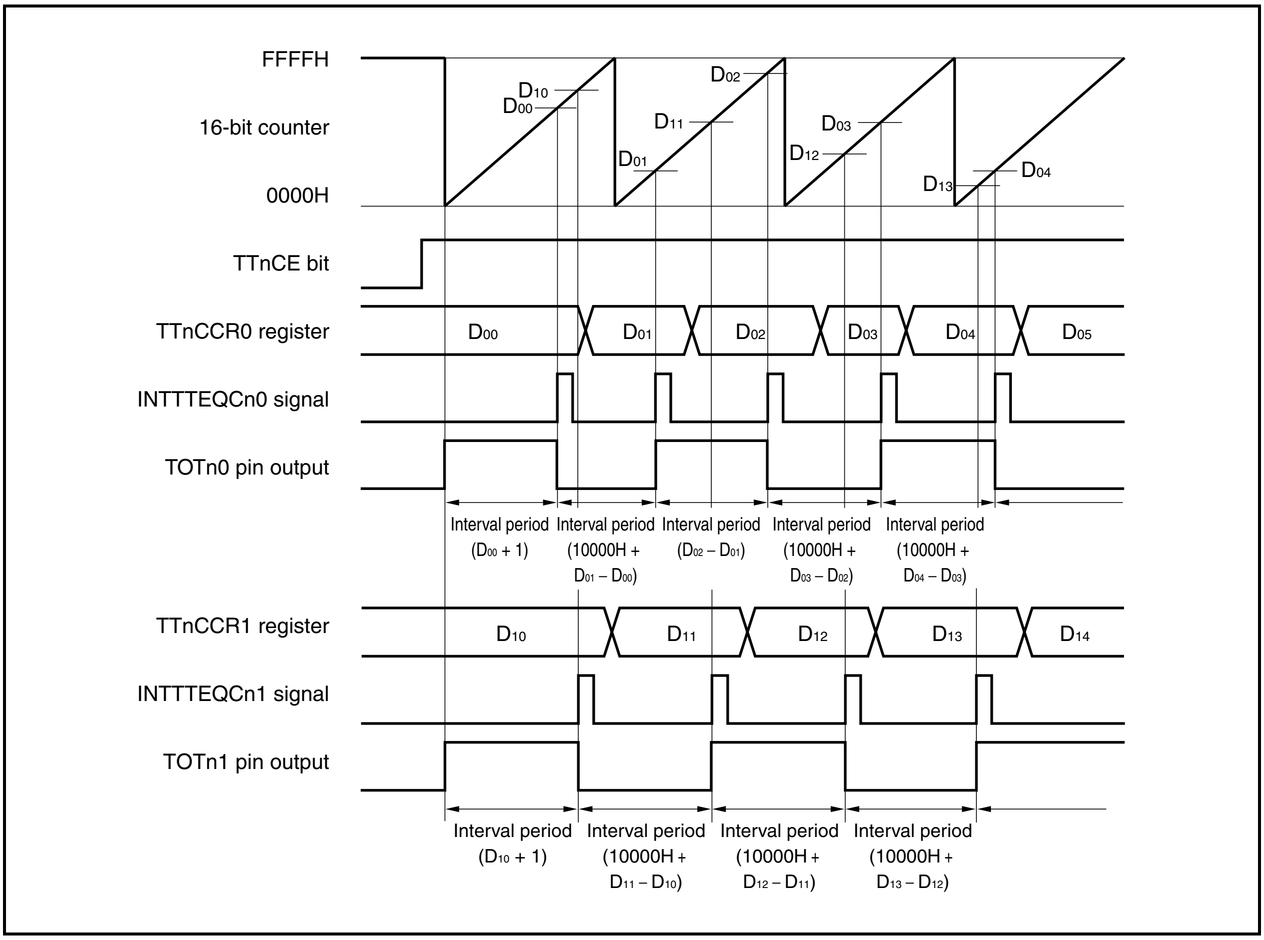
Counter is initialized and counting is stopped by clearing TTnCE bit to 0.

Remark n = 0 to 3

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter T is used as an interval timer with the TTnCCRa register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTTEQCna signal has been detected.


FFFFH
16-bit counter
0000H
D10
D00
D01
D11
D02
D12
D03
D13
D04
TTnCE bit
TTnCCR0 register
D00
D01
D02
D03
D04
D05
INTTTEQCn0 signal
TOTn0 pin output
Interval period (D00 + 1)
Interval period (10000H + D01 – D00)
Interval period (D02 – D01)
Interval period (10000H + D03 – D02)
Interval period (10000H + D04 – D03)
TTnCCR1 register
D10
D11
D12
D13
D14
INTTTEQCn1 signal
TOTn1 pin output
Interval period (D10 + 1)
Interval period (10000H + D11 – D10)
Interval period (10000H + D12 – D11)
Interval period (10000H + D13 – D12)

When performing an interval operation in the free-running timer mode, two intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TTnCCRa register must be re-set in the interrupt servicing that is executed when the INTTTEQCna signal is detected.

The set value for re-setting the TTnCCRa register can be calculated by the following expression, where "D_a" is the interval period.

Compare register default value: $D_a - 1$

Value set to compare register second and subsequent time: Previous set value + D_a

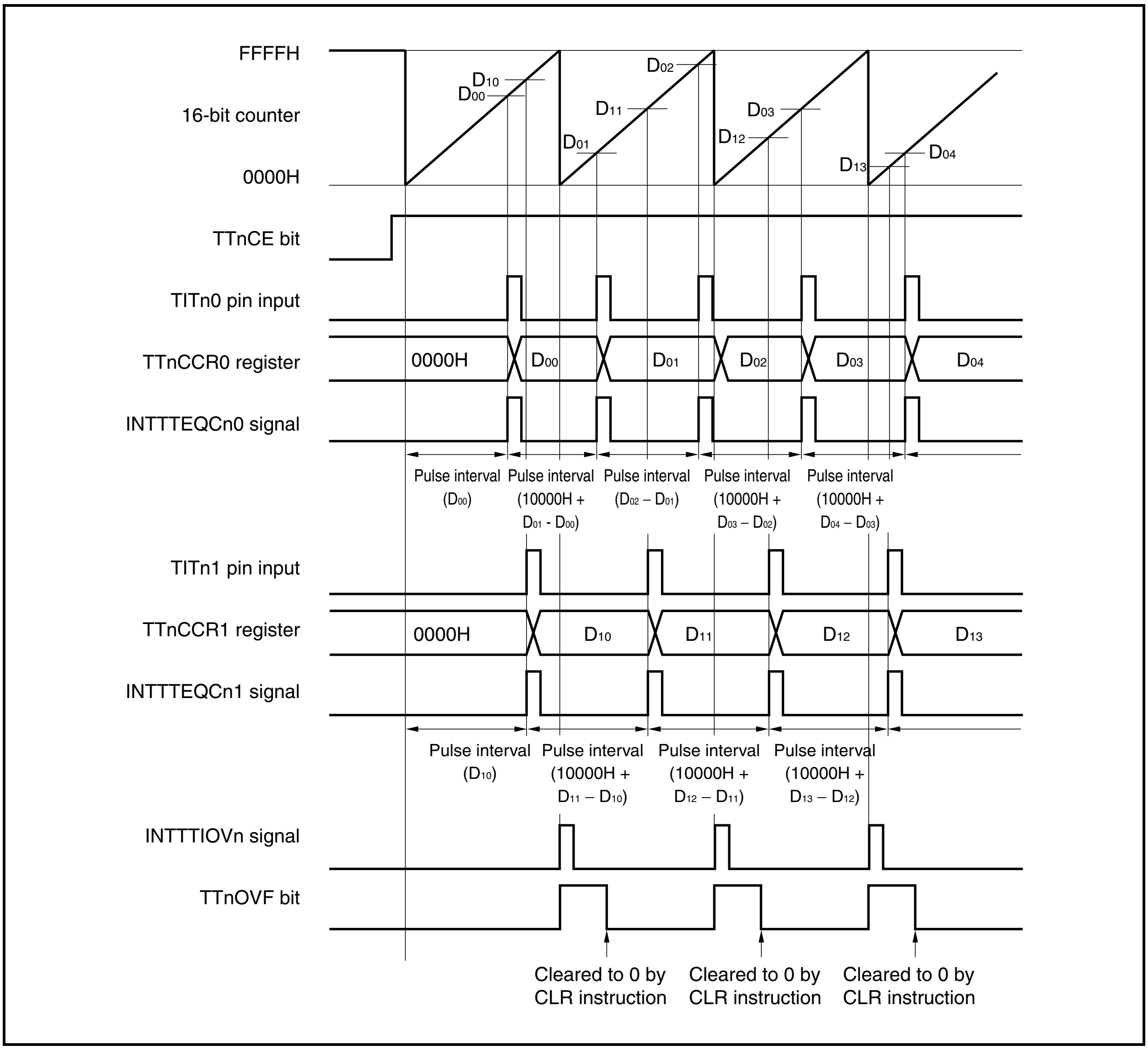
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0 to 3

a = 0, 1

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TTnCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTTEQCna signal has been detected and for calculating an interval.


FFFFH
16-bit counter
0000H
D_{10}
D_{00}
D_{02}
D_{11}
D_{01}
D_{03}
D_{12}
D_{13}
D_{04}
TTnCE bit
TITn0 pin input
TTnCCR0 register
0000H
D_{00}
D_{01}
D_{02}
D_{03}
D_{04}
INTTTEQCn0 signal
Pulse interval (D_{00})
Pulse interval (10000H + D_{01} - D_{00})
Pulse interval (D_{02} – D_{01})
Pulse interval (10000H + D_{03} – D_{02})
Pulse interval (10000H + D_{04} – D_{03})
TITn1 pin input
TTnCCR1 register
0000H
D_{10}
D_{11}
D_{12}
D_{13}
INTTTEQCn1 signal
Pulse interval (D_{10})
Pulse interval (10000H + D_{11} – D_{10})
Pulse interval (10000H + D_{12} – D_{11})
Pulse interval (10000H + D_{13} – D_{12})
INTTTIOVn signal
TTnOVF bit
Cleared to 0 by CLR instruction
Cleared to 0 by CLR instruction
Cleared to 0 by CLR instruction

When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

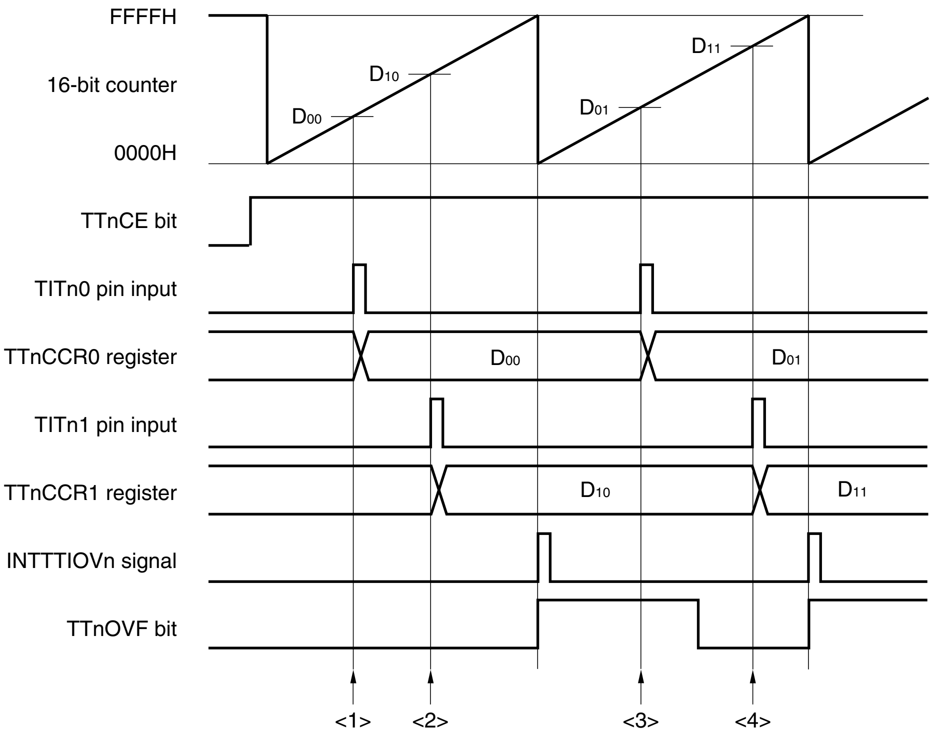
To measure a pulse width, the pulse width can be calculated by reading the value of the TTnCCRa register in synchronization with the INTTTEQCna signal, and calculating the difference between the read value and the previously read value.

Remark n = 0 to 3
a = 0, 1

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TTnCCR0 register (setting of the default value of the TITn0 pin input).
<2> Read the TTnCCR1 register (setting of the default value of the TITn1 pin input).
<3> Read the TTnCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<4> Read the TTnCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} − D_{10}) (incorrect).

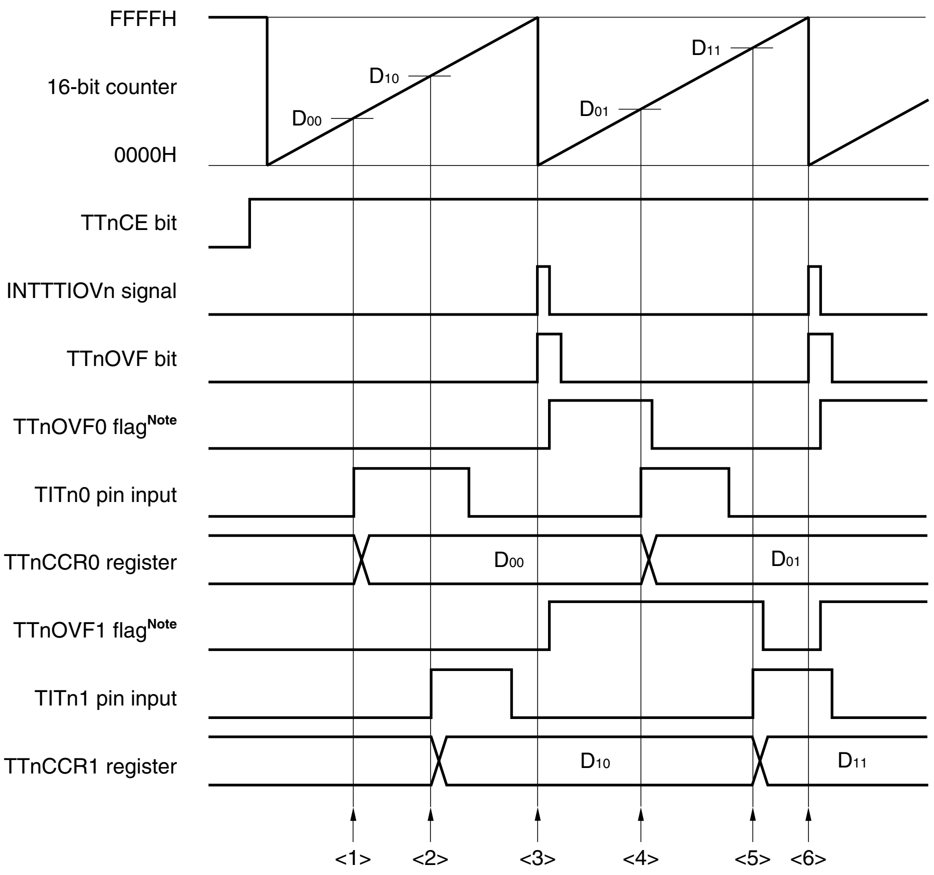
Remark n = 0 to 3

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.

Use software when using two capture registers. An example of how to use software is shown below.

(1/2)

Example when two capture registers are used (using overflow interrupt)



Note The TTnOVF0 and TTnOVF1 flags are set on the internal RAM by software.

<1> Read the TTnCCR0 register (setting of the default value of the TITn0 pin input).

<2> Read the TTnCCR1 register (setting of the default value of the TITn1 pin input).

<3> An overflow occurs. Set the TTnOVF0 and TTnOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

<4> Read the TTnCCR0 register.
Read the TTnOVF0 flag. If the TTnOVF0 flag is 1, clear it to 0.
Because the TTnOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).

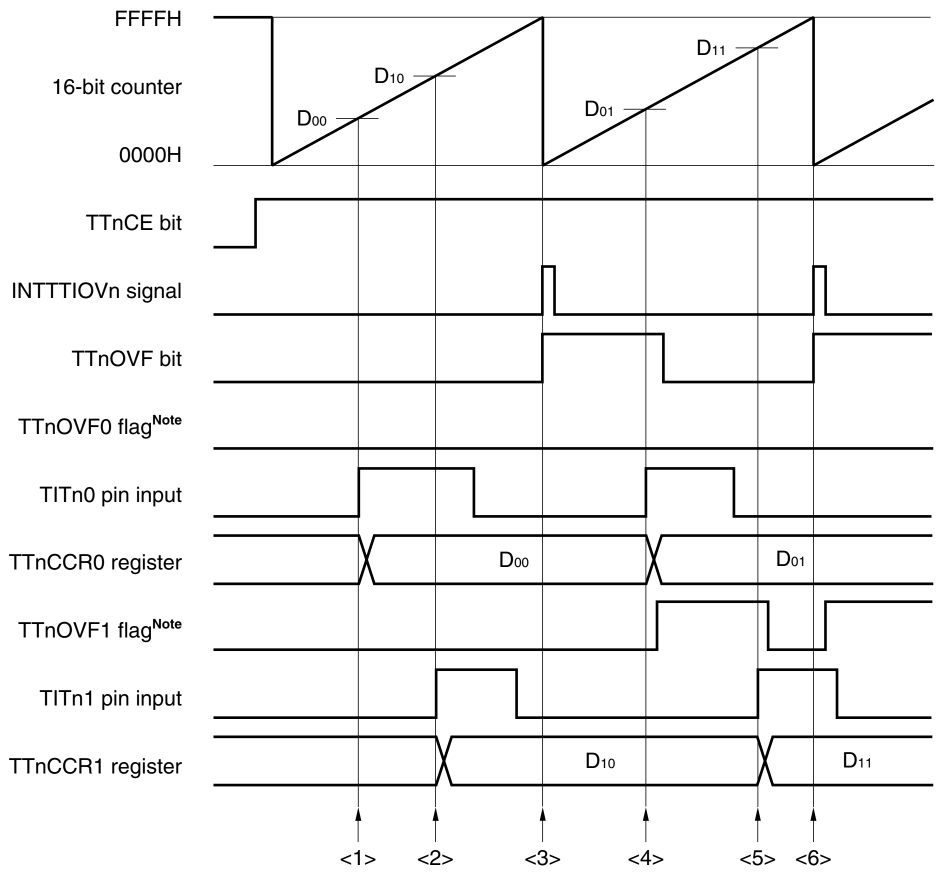
<5> Read the TTnCCR1 register.
Read the TTnOVF1 flag. If the TTnOVF1 flag is 1, clear it to 0 (the TTnOVF0 flag is cleared in <4>, and the TTnOVF1 flag remains 1).
Because the TTnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).

<6> Same as <3>

Remark n = 0 to 3

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TTnOVF0 and TTnOVF1 flags are set on the internal RAM by software.

<1> Read the TTnCCR0 register (setting of the default value of the TITn0 pin input).

<2> Read the TTnCCR1 register (setting of the default value of the TITn1 pin input).

<3> An overflow occurs. Nothing is done by software.

<4> Read the TTnCCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TTnOVF1 flag to 1, and clear the overflow flag to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).

<5> Read the TTnCCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TTnOVF1 flag. If the TTnOVF1 flag is 1, clear it to 0.
Because the TTnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).

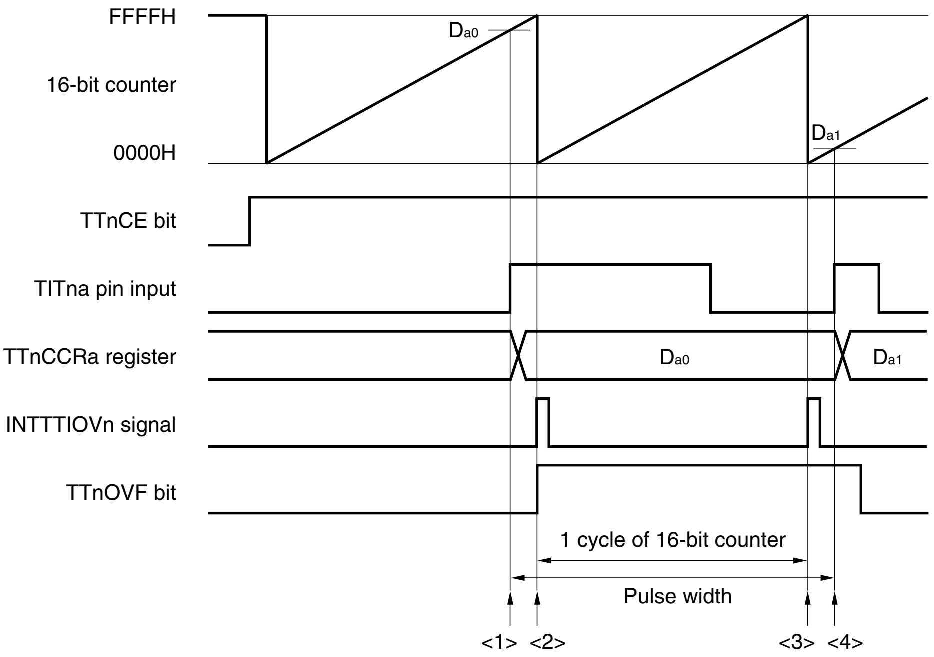
<6> Same as <3>

Remark n = 0 to 3

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



The following problem may occur when long pulse width is measured in the free-running timer mode.

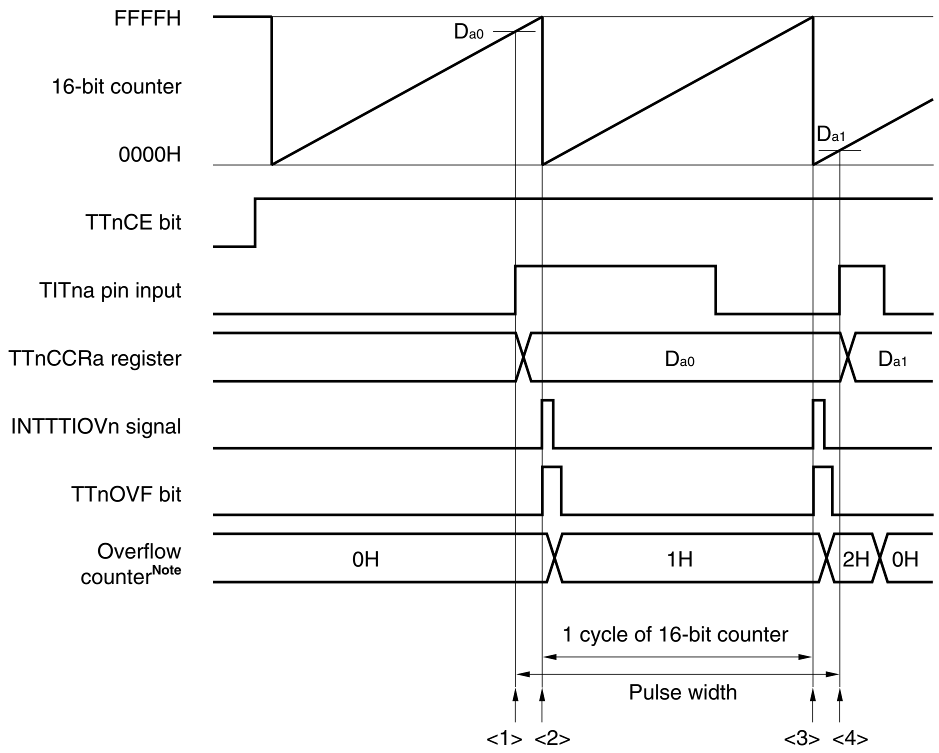
<1> Read the TTnCCRa register (setting of the default value of the TITna pin input).
<2> An overflow occurs. Nothing is done by software.
<3> An overflow occurs a second time. Nothing is done by software.
<4> Read the TTnCCRa register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{a1} − D_{a0}) (incorrect).
Actually, the pulse width must be (20000H + D_{a1} − D_{a0}) because an overflow occurs twice.

Remark n = 0 to 3
a = 0, 1

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.

If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TTnCCRa register (setting of the default value of the TITna pin input).

<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<3> An overflow occurs a second time. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<4> Read the TTnCCRa register.
Read the overflow counter.
→ When the overflow counter is "N", the pulse width can be calculated by (N × 10000H + D_{a1} – D_{a0}).
In this example, the pulse width is (20000H + D_{a1} – D_{a0}) because an overflow occurs twice.
Clear the overflow counter (0H).

Remark n = 0 to 3
a = 0, 1

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TTnOVF bit to 0 with the CLR instruction after reading the TTnOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TTnOPT0 register after reading the TTnOVF bit when it is 1.

8.6.7 Pulse width measurement mode (TTnMD3 to TTnMD0 bits = 0110)

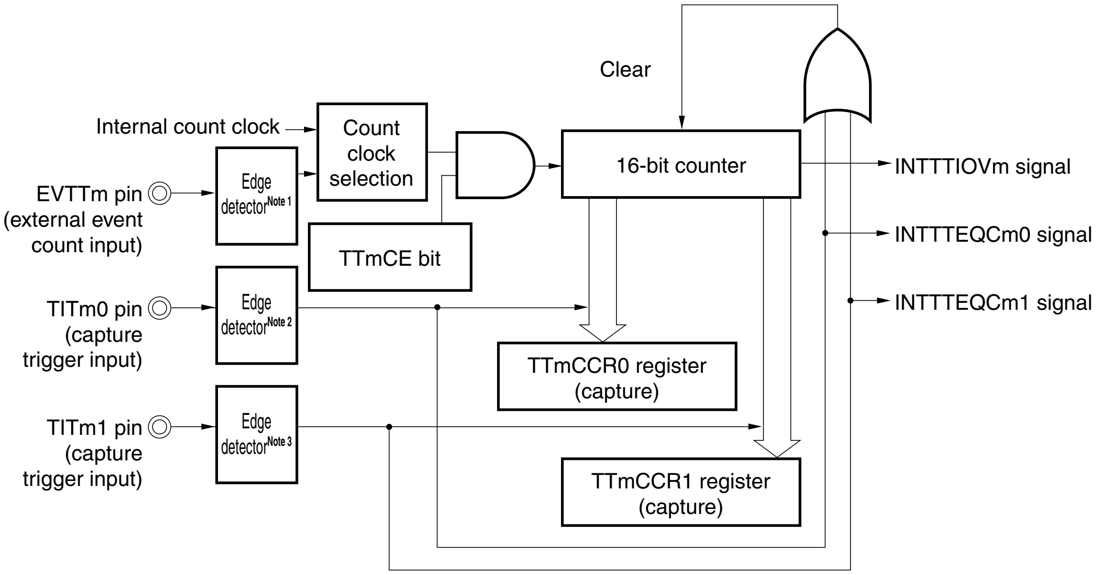
In the pulse width measurement mode, 16-bit timer/event counter T starts counting when the TTnCTL0.TTnCE bit is set to 1. Each time the valid edge input to the TITna pin has been detected, the count value of the 16-bit counter is stored in the TTnCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TTnCCRa register after a capture interrupt request signal (INTTTEQCna) occurs.

As shown in Figure 8-46, select either the TITn0 or TITn1 pin as the capture trigger input pin and set the unused pins to “No edge detection” by using the TTnIOC1 register.

Remark n = 0 to 3
a = 0, 1

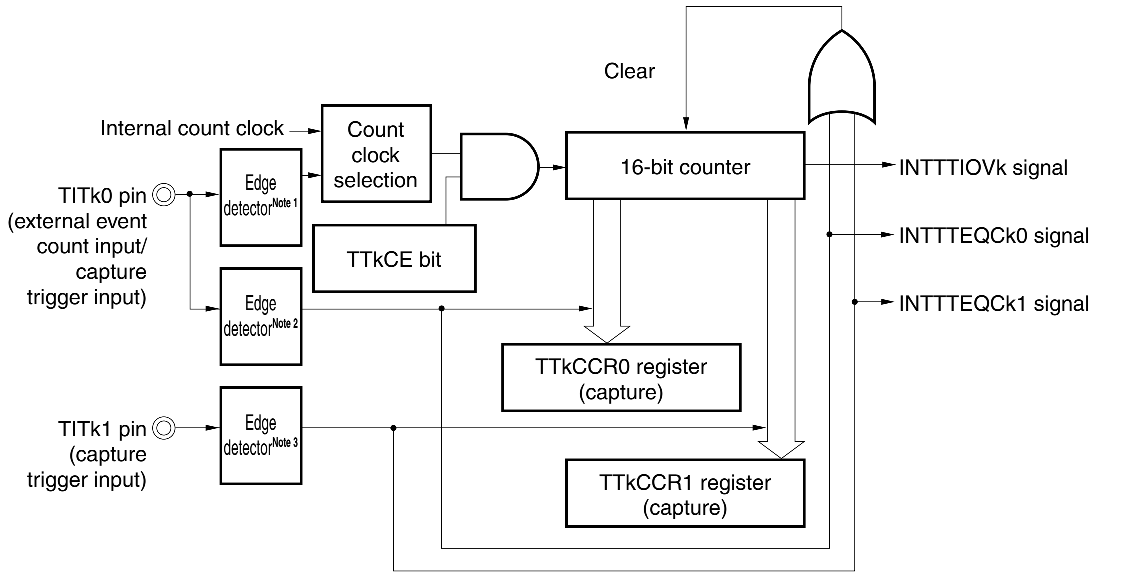
Figure 8-44. Configuration of TMT0 and TMT1 in Pulse Width Measurement Mode



Notes 1. Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.
2. Set by the TTmIOC1.TTmIS1 and TTmIOC1.TTmIS0 bits.
3. Set by the TTmIOC1.TTmIS3 and TTmIOC1.TTmIS2 bits.

Remark m = 0, 1

Figure 8-45. Configuration of TMT2 and TMT3 in Pulse Width Measurement Mode



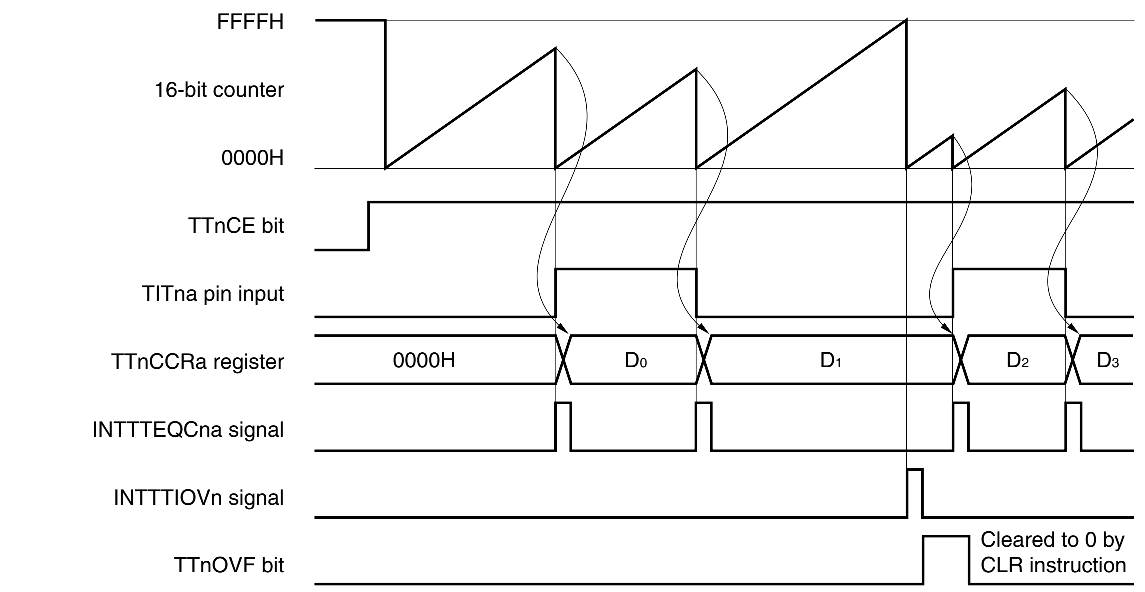
Notes 1. Set by the TTkIOC2.TTkEES1 and TTkIOC2.TTkEES0 bits.

2. Set by the TTkIOC1.TTkIS1 and TTkIOC1.TTkIS0 bits.

3. Set by the TTkIOC1.TTkIS3 and TTkIOC1.TTkIS2 bits.

Remark k = 2, 3

Figure 8-46. Basic Timing in Pulse Width Measurement Mode



Remark n = 0 to 3
a = 0, 1

When the TTnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TITna pin is later detected, the count value of the 16-bit counter is stored in the TTnCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTTEQCna) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

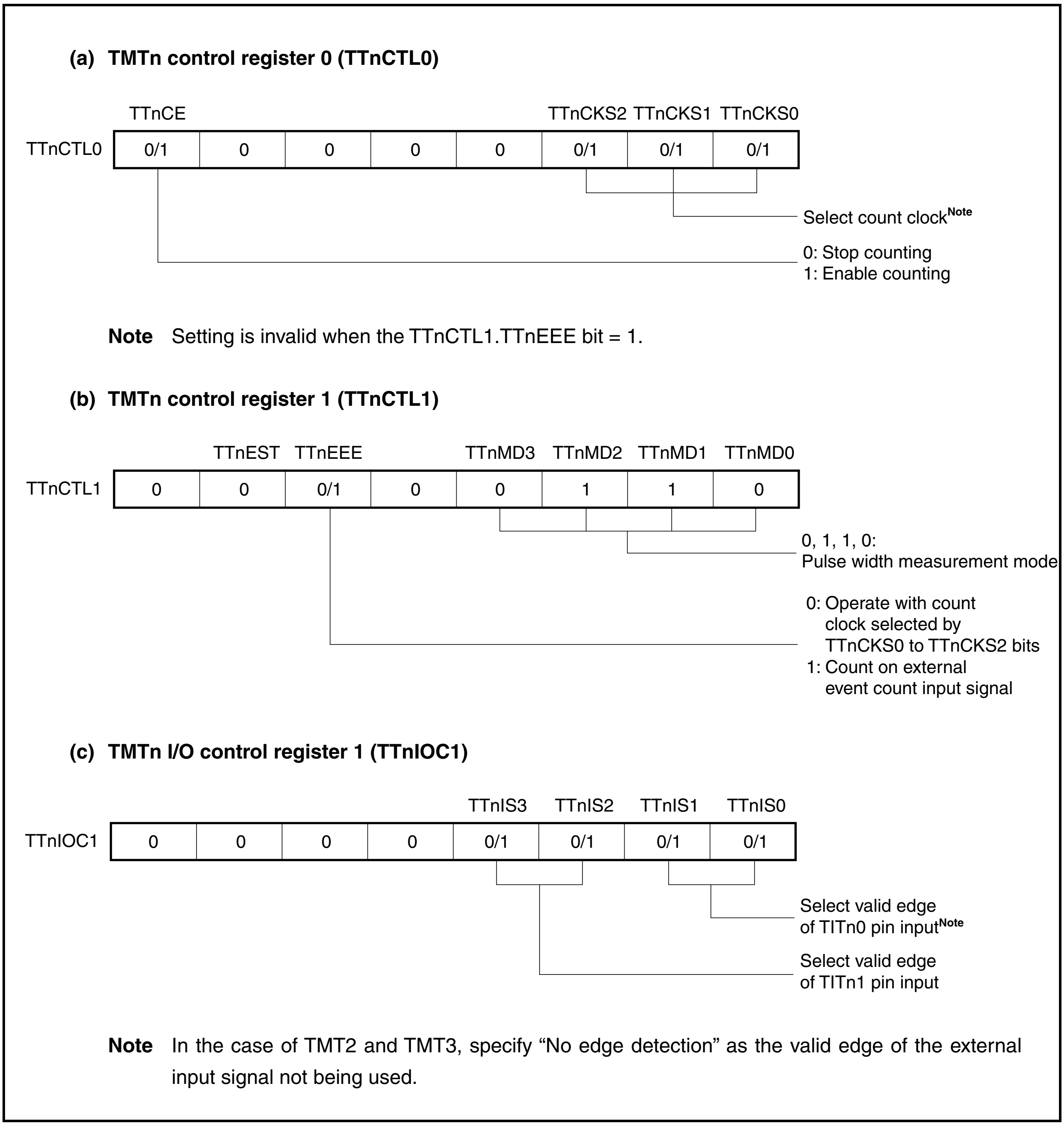
If the valid edge is not input to the TITma pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTTIOVn) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TTnOPT0.TTnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TTnOVF bit set (1) count + Captured value) × Count clock cycle

Remark n = 0 to 3
a = 0, 1

Figure 8-47. Register Setting in Pulse Width Measurement Mode (1/2)



(a) TMTn control register 0 (TTnCTL0)

TTnCE
TTnCKS2 TTnCKS1 TTnCKS0
TTnCTL0
0/1 0 0 0 0 0/1 0/1 0/1
Select count clockNote
0: Stop counting
1: Enable counting

Note Setting is invalid when the TTnCTL1.TTnEEE bit = 1.

(b) TMTn control register 1 (TTnCTL1)

TTnEST TTnEEE
TTnMD3 TTnMD2 TTnMD1 TTnMD0
TTnCTL1
0 0 0/1 0 0 1 1 0
0, 1, 1, 0:
Pulse width measurement mode
0: Operate with count clock selected by TTnCKS0 to TTnCKS2 bits
1: Count on external event count input signal

(c) TMTn I/O control register 1 (TTnIOC1)

TTnIS3 TTnIS2 TTnIS1 TTnIS0
TTnIOC1
0 0 0 0 0/1 0/1 0/1 0/1
Select valid edge of TITn0 pin inputNote
Select valid edge of TITn1 pin input

Note In the case of TMT2 and TMT3, specify “No edge detection” as the valid edge of the external input signal not being used.

Figure 8-47. Register Setting in Pulse Width Measurement Mode (2/2)

(d) TMTn I/O control register 2 (TTnIOC2)

					TTnEES1	TTnEES0	TTnETS1	TTnETS0
TTnIOC2	0	0	0	0	0/1	0/1	0	0

TTnEES1, TTnEES0: Select valid edge of external event count input[Notes 1, 2]

Notes 1. TMT0 and TMT1: EVTTm pin
TMT2 and TMT3: TITk0 pin

2. In the case of TMT2 and TMT3, specify "No edge detection" as the valid edge of the external input signal not being used.

(e) TMTn option register 0 (TTnOPT0)

			TTnCCS1	TTnCCS0				TTnOVF
TTnOPT0	0	0	0	0	0	0	0	0/1

TTnOVF: Overflow flag

(f) TMTn counter read buffer register (TTnCNT)

The value of the 16-bit counter can be read by reading the TTnCNT register.

(g) TMTn capture/compare registers 0 and 1 (TTnCCR0 and TTnCCR1)

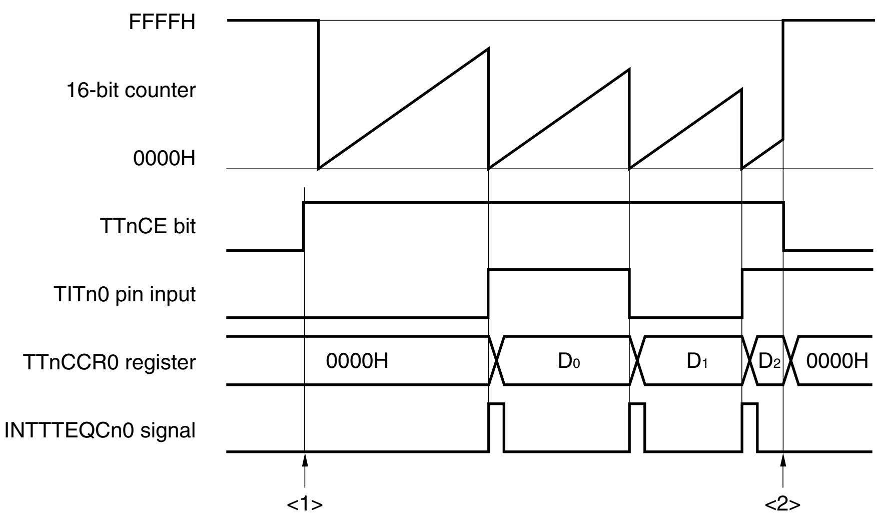
These registers store the count value of the 16-bit counter when the valid edge input to the TITn0 and TITn1 pins is detected.

Remarks 1. TMTm control register 2 (TTmCTL2), TMTn I/O control register 0 (TTnIOC0), TMTm I/O control register 3 (TTmIOC3), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the pulse width measurement mode.

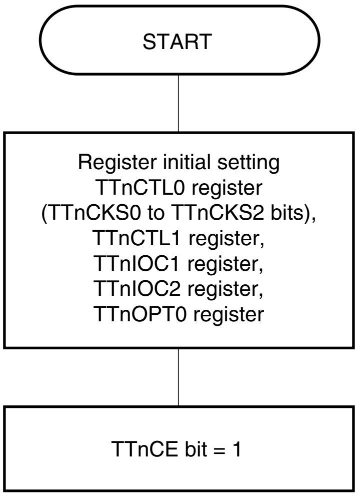
2. n = 0 to 3
m = 0, 1
k = 2, 3

(1) Operation flow in pulse width measurement mode

Figure 8-48. Software Processing Flow in Pulse Width Measurement Mode



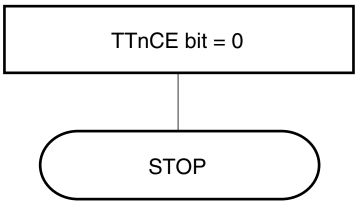
<1> Count operation start flow



Initial setting of these registers is performed before setting the TTnCE bit to 1.

The TTnCKS0 to TTnCKS2 bits can be set at the same time as when counting starts (TTnCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TTnCE bit to 0.

Remark n = 0 to 3

(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

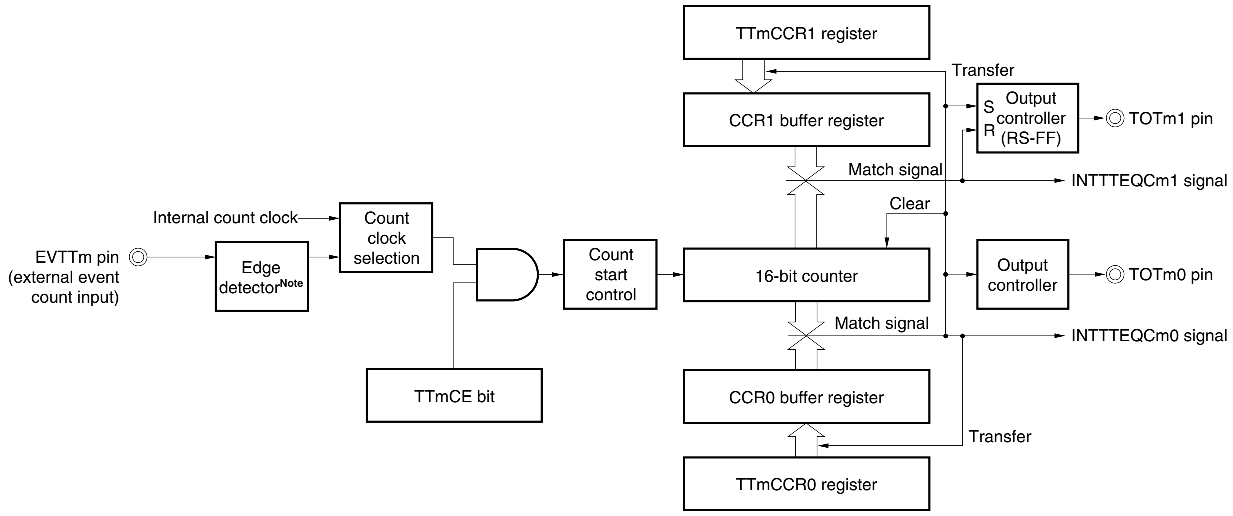
The overflow flag can be cleared to 0 by clearing the TTnOVF bit to 0 with the CLR instruction after reading the TTnOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TTnOPT0 register after reading the TTnOVF bit when it is 1.

8.6.8 Triangular-wave PWM output mode (TTnMD3 to TTnMD0 bits = 0111)

In the triangular-wave PWM output mode, a triangular-wave PWM waveform is output from the TOTn1 pin when the TTnCTL0.TTnCE bit is set to 1.

An inverted PWM waveform is output from the TOTn0 pin when the count value of the 16-bit counter matches the value of the CCR0 buffer register and when the 16-bit counter is set to 0000H.

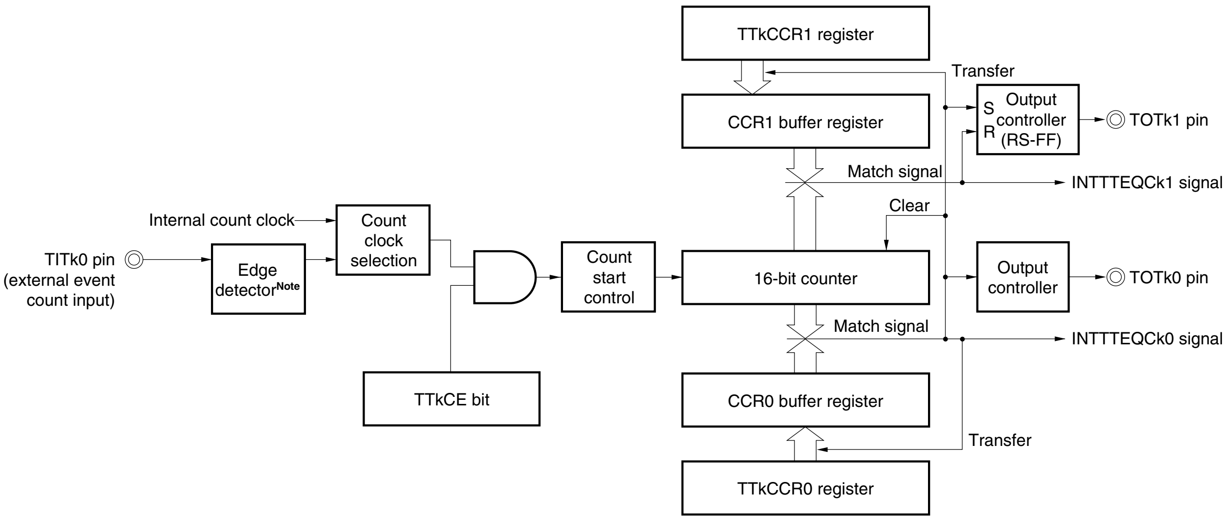
Figure 8-49. Configuration of TMT0 and TMT1 in Triangular-Wave PWM Output Mode



Note Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark m = 0 to 3

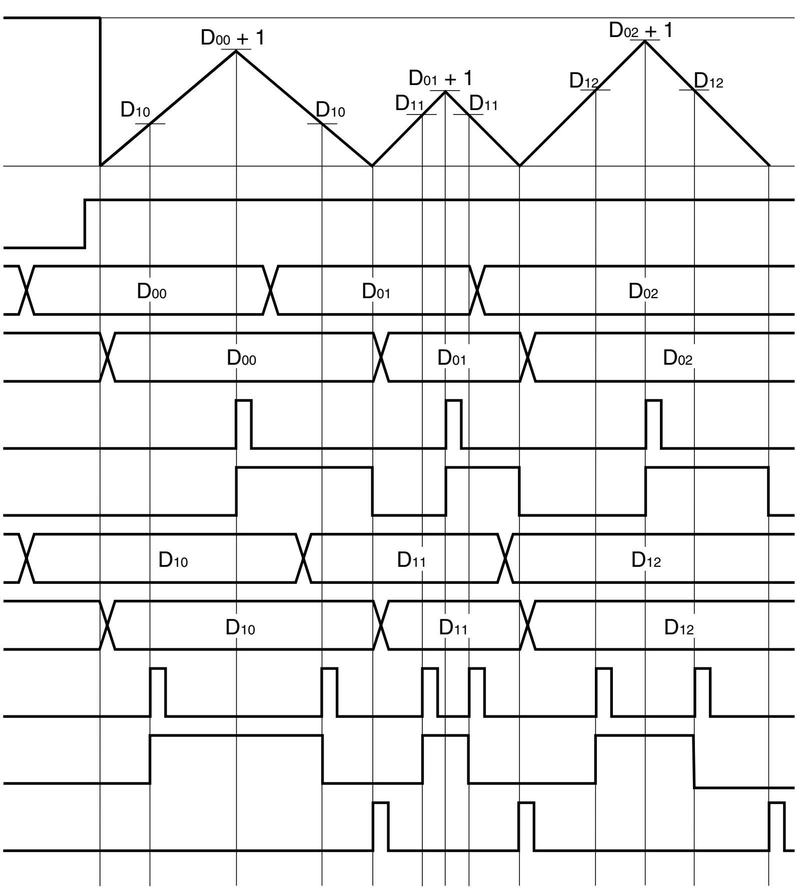
Figure 8-50. Configuration of TMT2 and TMT3 in Triangular-Wave PWM Output Mode



Note Set by the TTkIOC2.TTkEES1 and TTkIOC2.TTkEES0 bits.

Remark k = 2, 3

Figure 8-51. Basic Timing in Triangular-Wave PWM Output Mode



Remarks 1. n = 0 to 3

2. Timing chart when TTnIOC0 register = 05H.

The 16-bit counter is cleared from FFFFH and 0000H and starts counting when the TTnCE bit is set to 1. The triangular PWM waveform is output from the TOTn1 pin.

In the triangular-wave PWM output mode, the counter counts up or down. When the 16-bit counter reaches 0000H while it is counting down, an overflow interrupt request signal (INTTTIOVn) is generated. At this time, the TTnOPT0.TTnOVF bit is not set to 1. If the count value of the 16-bit counter matches the value of the CCR0 buffer register while the counter is counting up, a compare match interrupt request signal (INTTTEQCn0) is generated.

The counting direction is changed from up to down when the value of the 16-bit counter matches that of the CCR0 buffer register, and from down to up when the counter is cleared to 0000H.

The PWM waveform can be changed by rewriting the TTnCCRa register during operation. To change the PWM waveform during operation, write the TTnCCR1 register last.

The cycle of the triangular PWM waveform is set by the TTnCCR0 register and its duty factor is set by the TTnCCR1 register. Set a value to the TTnCCR0 register in a range of “0 ≤ TTnCCR0 ≤ FFFEH”. The rewritten value is reflected when the 16-bit counter reaches 0000H while it is counting down.

Even when changing only the cycle of the PWM waveform, first set a period to the TTnCCR0 register, and then write the same value (value same as that set to the TTnCCR1 register) to the TTnCCR1 register.

To transfer data from the TTnCCRa register to the CCRa buffer register, the data must be written to the TTnCCR1 register (a = 0, 1).

(1) PWM output of 0%/100%

In the triangular-wave PWM output mode, 0% waveform output and 100% waveform output are available for PWM output.

The 0% waveform is output by setting the TTnCCR1 register to “M + 1” when the TTnCCR0 register = M.

The 100% waveform is output by setting the TTnCCR1 register to “0000H”.

The output level of TOTn0 and TOTn1 can be set in the TTnIOC0 register.

Remark n = 0 to 3

Figure 8-52. 0% PWM Output Waveform (TTnIOC0 Register = 05H)

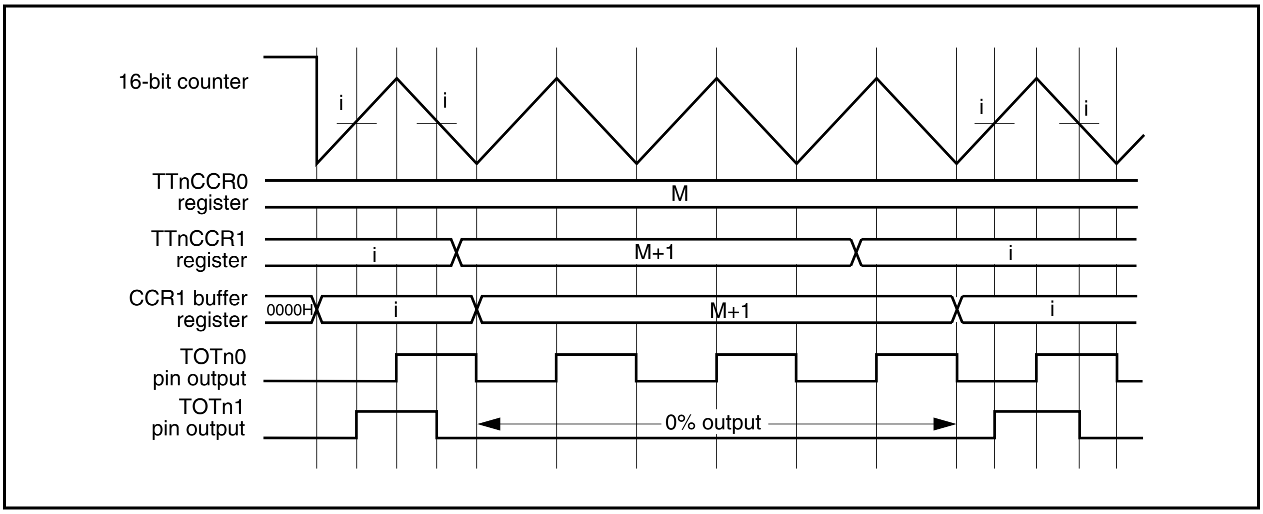
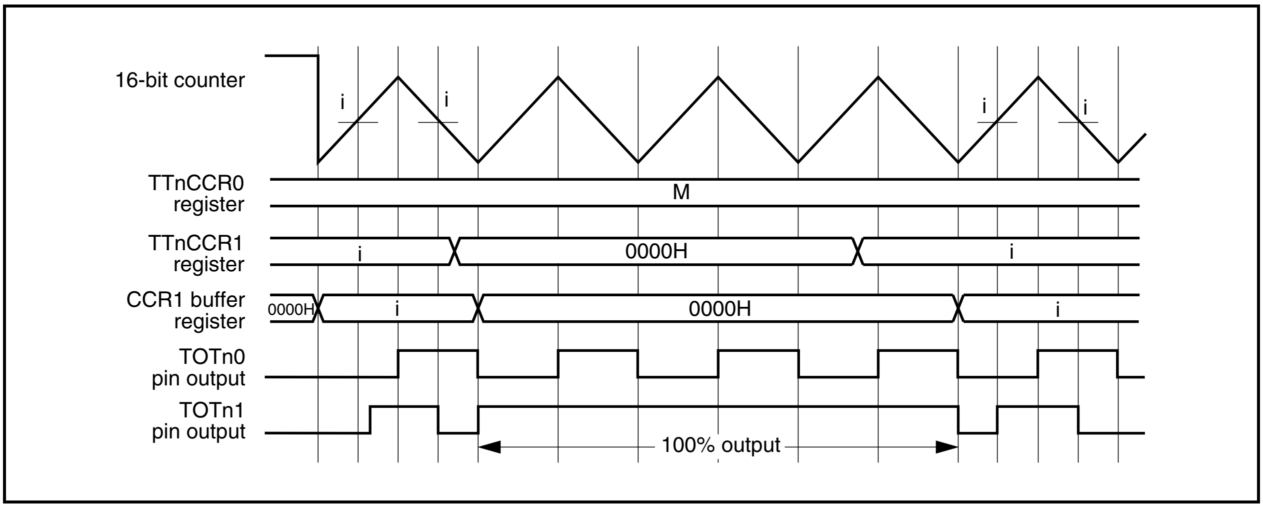


Figure 8-53. 100% PWM Output Waveform (TTnIOC0 Register = 05H)



8.6.9 Encoder count function

The encoder count function includes an encoder compare mode (see **8.6.10 Encoder compare mode (TTmMD3 to TTmMD0 bits = 1000)**).

Mode	TTmCCR0 Register	TTmCCR1 Register
Encoder compare mode	Compare only	Compare only

(1) Count-up/-down control

Counting up or down by the 16-bit counter is controlled by the phase of input encoder signals (TENCm0 and TENCm1) and setting of the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits.

When the encoder count function is used, the internal count clock and external event count input (EVTTm) cannot be used. Set the TTmCTL0.TTmCKS2 to TTmCTL0.TTmCKS0 bits to 000 and the TTmCTL1.TTmEEE bit to 0.

(2) Setting initial value of 16-bit counter

The initial count value set to the TTmTCW register when the TTmCTL2.TTmECC bit = 0 is transferred to the 16-bit counter immediately after the counter starts its operation (TTmCTL0.TTmCE bit = 0 → 1), and the counter starts the operation after it detects the valid edge of the encoder input signal (TENCm0 or TENCm1).

(3) Basic operation

The TTmCCRa register generates a compare match interrupt request signal (INTTTEQCma) when the count value of the 16-bit counter matches the value of the CCRa buffer register.

(4) Clear operation

The 16-bit counter is cleared when the following conditions are satisfied in the encoder compare mode.

- When the value of the 16-bit counter matches the value of the compare register (the TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits are set)
- When the edge of the encoder clear input signal (TECRm) is detected and cleared (the TTmECS1 and TTmECS0 bits are set when the TTmIOC3.TTmSCE bit = 0)
- When the clear level condition of the TENCm0, TENCm1, and TECRm pins is detected (the TTmZCL, TTmBCL, and TTmACL bits are set when the TTmSCE bit = 1)

Remark m = 0, 1
a = 0, 1

(5) Controlling bits of TTmCTL2 register

The setting of the TTmCTL2 register in the encoder compare mode is shown below.

Table 8-9. Setting of TTmCTL2 Register

Mode	TTmUDS1, TTmUDS0 Bits (<1>)	TTmECM1 Bit (<2>)	TTmECM0 Bit (<2>)	TTmLDE Bit (<3>)	Counter Clear (Target Compare Register)	Transfer to Counter
Encoder compare mode	Can be set to 00, 01, 10, or 11.	0	0	0	–	–
				1		Possible
			1	0	TTmCCR0	–
				1		Possible[Note]
		1	0	Invalid	TTmCCR1	–
			1	Invalid	TTmCCR0, TTmCCR1	–

Note The counter can operate in a range from 0000H to the set value of the TTmCCR0 register.

Remark m = 0, 1

(a) Outline of each bit

<1> The TTmUDS1 and TTmUDS0 bits identify the counting direction (up or down) of the 16-bit counter by the phase input from the encoder input pin (TENCm0 or TENCm1).

<2> The TTmECM1 and TTmECM0 bits control clearing of the 16-bit counter when its count value matches the value of the CCR0 or CCR1 buffer register.

<3> The TTmLDE bit controls a function to transfer the set value of the TTmCCR0 register to the 16-bit counter when the counter underflows. The TTmLDE bit is valid only when the TTmECM1 and TTmECM0 bits are 00 or 01. It is invalid when these bits are set to any other value.

(b) Detailed explanation of each bit

<1> TTmUDS1 and TTmUDS0 bits: Count-up/-down selection

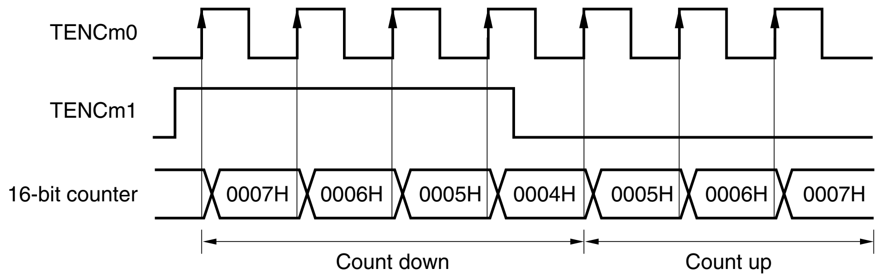
Whether the 16-bit counter is counting up or down is identified by the phase input from the TENCm0 or TENCm1 pin and depending on the setting of the TTmUDS1 and TTmUDS0 bits. These bits are valid only in the encoder compare mode.

- When TTmUDS1 and TTmUDS0 bits = 00

TENCm0 Pin	TENCm1 Pin	Count Operation
Rising edge	High level	Count down
Falling edge		
Both edges		
Rising edge	Low level	Count up
Falling edge		
Both edges		

Remark Detecting the edge of the TENCm0 pin is specified by the TTmIOC3.TTmEIS1 and TTmEIS0 bits.

Figure 8-54. Operation Example (When Valid Edge of TENCm0 Pin Is Specified to Be Rising Edge and No Edge Is Specified as Valid Edge of TENCm1 Pin)



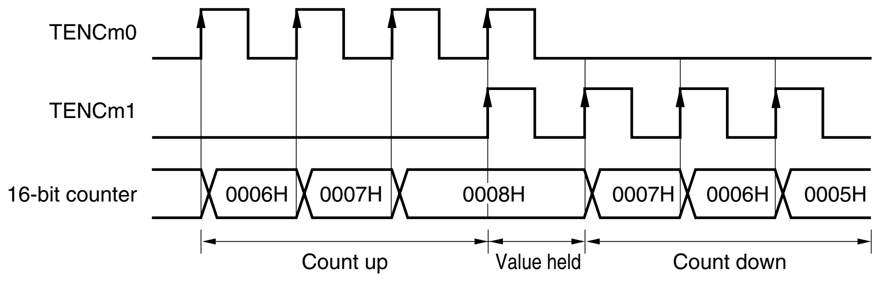
Remark m = 0, 1

- When TTmUDS1 and TTmUDS0 bits = 01

TENCm0 Pin	TENCm1 Pin	Count Operation
Low level	Rising edge	Count down
	Falling edge	
	Both edges	
High level	Rising edge	
	Falling edge	
	Both edges	
Rising edge	High level	Count up
Falling edge		
Both edges		
Rising edge	Low level	
Falling edge		
Both edges		
Simultaneous input to TENCm0 and TENCm1 pins		Counter does not perform count operation but holds value immediately before.

Remark Detecting the edge of the TENCm0 and TENCm1 pins is specified by the TTmIOC3.TTmEIS1 and TTmIOC3.TTmEIS0 bits.

Figure 8-55. Operation Example (When Rising Edge Is Specified as Valid Edge of TENCm0 and TENCm1 Pins)



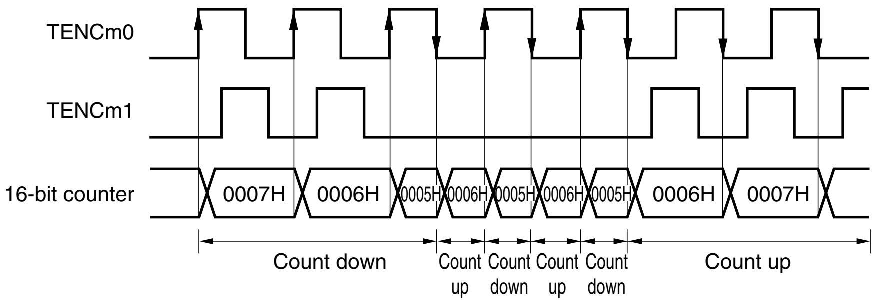
Remark m = 0, 1

- When TTmUDS1 and TTmUDS0 bits = 10

TENCm0 Pin	TENCm1 Pin	Count Operation
Low level	Falling edge	Counter does not perform count operation but holds value immediately before.
Rising edge	Low level	Count down
High level	Rising edge	Counter does not perform count operation but holds value immediately before.
Falling edge	High level	
Rising edge		
High level	Falling edge	
Falling edge	Low level	Count up
Low level	Rising edge	Counter does not perform count operation but holds value immediately before.
Rising edge		
Falling edge		
Rising edge	Falling edge	Count down
Falling edge		Count up

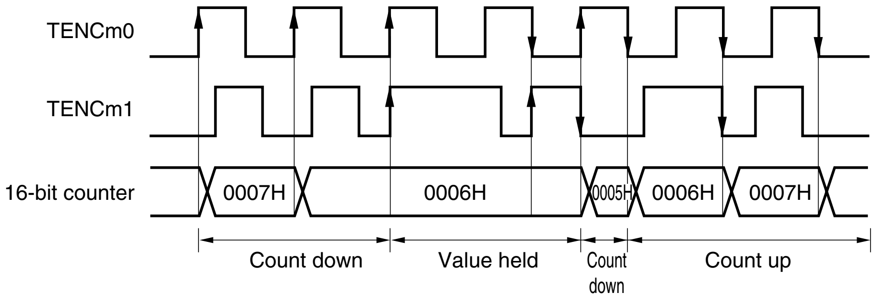
Caution Specification of the valid edge of the TENCm0 and TENCm1 pins is invalid.

Figure 8-56. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins Do Not Overlap)



Remark m = 0, 1

Figure 8-57. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins Overlap)



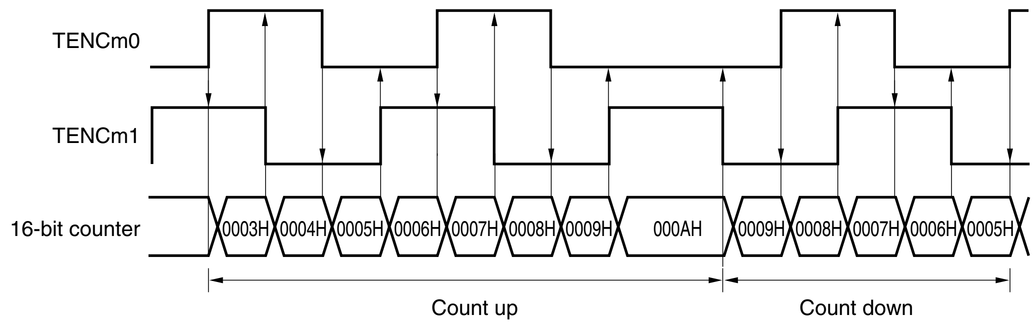
Remark m = 0, 1

- When TTmUDS1 and TTmUDS0 bits = 11

TENCm0 Pin	TENCm1 Pin	Count Operation
Low level	Falling edge	Count down
Rising edge	Low level	
High level	Rising edge	
Falling edge	High level	
Rising edge		Count up
High level	Falling edge	
Falling edge	Low level	
Low level	Rising edge	
Simultaneous input to TENCm0 and TENCm1 pins		Counter does not perform count operation but holds value immediately before.

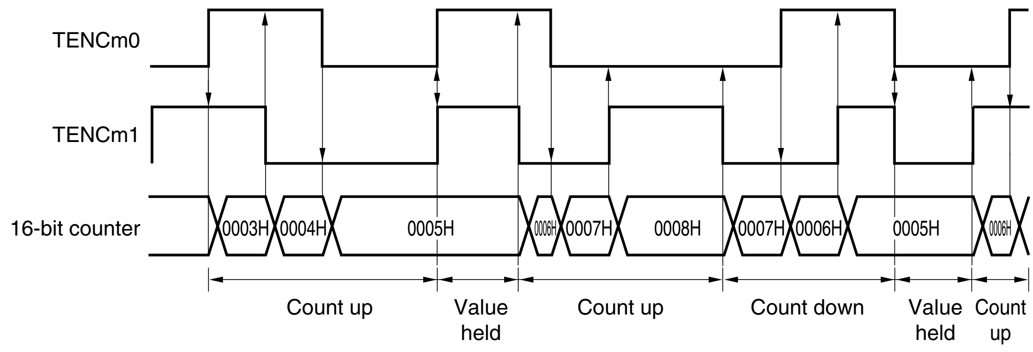
Caution Specification of the valid edge of the TENCm0 and TENCm1 pins is invalid.

Figure 8-58. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins Do Not Overlap)



Remark m = 0, 1

Figure 8-59. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins Overlap)



Remark m = 0, 1

<2> TTmECM1 and TTmECM0 bits: Timer/counter clear function upon match of the compare register

The 16-bit counter performs its count operation in accordance with the set value of the TTmECM1 and TTmECM0 bits when the count value of the counter matches the value of the CCRa buffer register.

- When TTmECM1 and TTmECM0 bits = 00

  The 16-bit counter is not cleared when its count value matches the value of the CCRa buffer register.

- When TTmECM1 and TTmECM0 bits = 01

  The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR0 buffer register.

Next Count Operation	Description
Count up	16-bit counter is cleared to 0000H.
Count down	Count value of 16-bit counter is counted down.

- When TTmECM1 and TTmECM0 bits = 10

  The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR1 buffer register.

Next Count Operation	Description
Count up	Count value of 16-bit counter is counted up.
Count down	16-bit counter is cleared to 0000H.

- When TTmECM1 and TTmECM0 bits = 11

  The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR0 buffer register.

Next Count Operation	Description
Count up	16-bit counter is cleared to 0000H.
Count down	Count value of 16-bit counter is counted down.

The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR1 buffer register.

Next Count Operation	Description
Count up	Count value of 16-bit counter is counted up.
Count down	16-bit counter is cleared to 0000H.

<3> TTmLDE bit: Transfer function of the set value of the TTmCCR0 register to the 16-bit counter when the counter underflows

When the TTmLDE bit = 1, the set value of the TTmCCR0 register can be transferred to the 16-bit counter when the counter underflows.

The TTmLDE bit is valid only in the encoder compare mode.

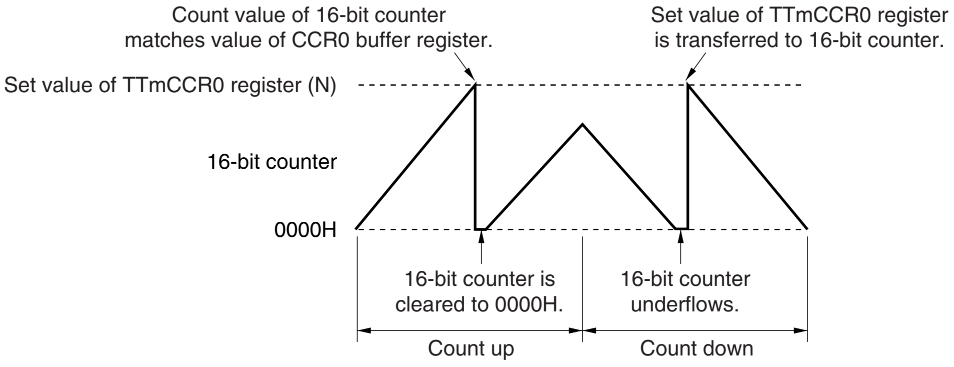
- Count operation in range from 0000H to set value of the TTmCCR0 register

  If the 16-bit counter performs a count operation when the TTmLDE bit = 1 and TTmECM1 and TTmECM0 bits = 01, and when the count value of the counter matches the set value of the CCR0 buffer register when the TTmECM0 bit = 1, the 16-bit counter is cleared to 0000H if the next count operation is counting up.

  If the 16-bit counter underflows when the TTmLDE bit = 1, the set value of the TTmCCR0 register is transferred to the counter.

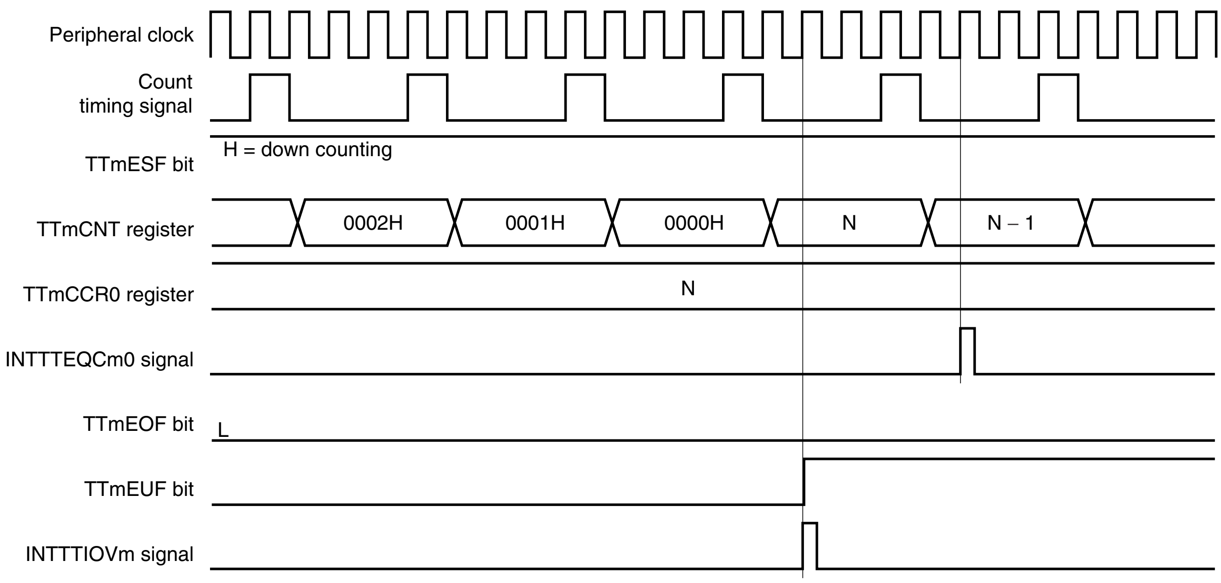
  Therefore, the counter can operate in a range from 0000H to the set value of the TTmCCR0 register in which the upper-limit count value is the set value of the TTmCCR0 register and the lower-limit value is 0000H.

Figure 8-60. Operation Example (Count Operation in Range from 0000H to Set Value of TTmCCR0 Register)


Count value of 16-bit counter
matches value of CCR0 buffer register.
Set value of TTmCCR0 register
is transferred to 16-bit counter.
Set value of TTmCCR0 register (N)
16-bit counter
0000H
16-bit counter is
cleared to 0000H.
16-bit counter
underflows.
Count up
Count down

Remark m = 0, 1

Figure 8-61. Operation Timing (Count Operation in Range from 0000H to Set Value of TTmCCR0 Register)



Remarks 1. TTmESF bit: Bit 0 of TMTm option register 1 (TTmOPT1)
TTmEOF bit: Bit 1 of TMTm option register 1 (TTmOPT1)
TTmEUF bit: Bit 2 of TMTm option register 1 (TTmOPT1)

2. m = 0, 1

(6) Clearing counter to 0000H by encoder clear signal (TECRm pin)

The 16-bit counter can be cleared to 0000H by the input signal of the TECRm pin in two ways which are selected by the TTmIOC3.TTmSCE bit. The TTmSCE bit also controls, depending its setting, the TTmIOC3.TTmZCL, TTmIOC3.TTmBCL, TTmIOC3.TTmACL, TTmIOC3.TTmESC1, and TTmIOC3.TTmECS0 bits.

The counter can be cleared by the methods described below only in the encoder compare mode.

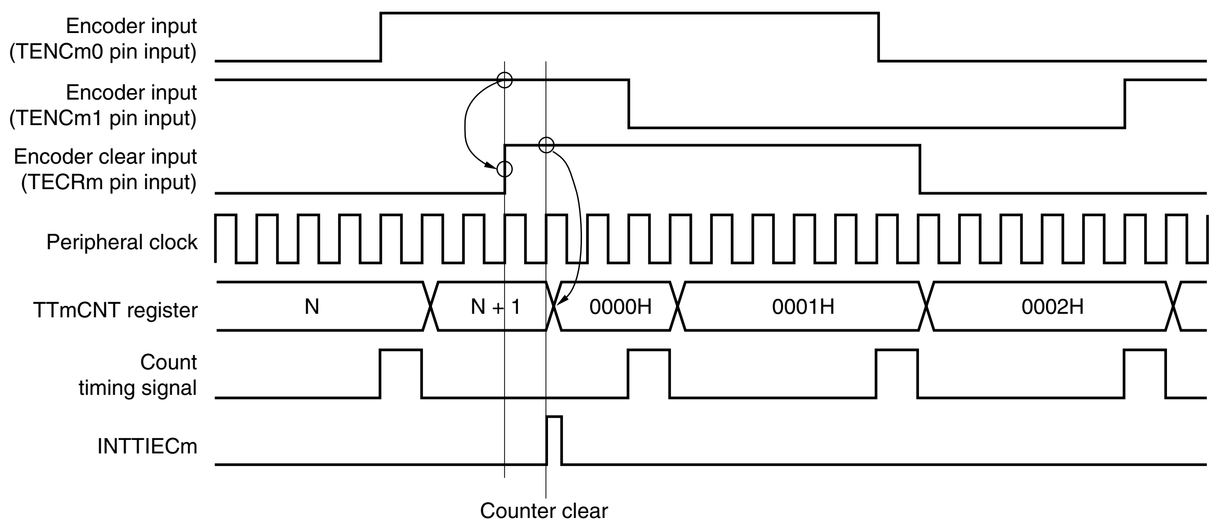
Table 8-10. Relationship Between TTmSCE Bit and TTmZCL, TTmBCL, TTmACL, TTmECS1, and TTmECS0 Bits

Clearing Method	TTmSCE Bit	TTmZCL Bit	TTmBCL Bit	TTmACL Bit	TTmECS1, TTmECS0 Bits
<1>	0	Invalid	Invalid	Invalid	Valid
<2>	1	Valid	Valid	Valid	Invalid

(a) Clearing method <1>: By detecting edge of encoder clear signal (TECRm pin) (TTmSCE bit = 0)

When the TTmSCE bit = 0, the 16-bit counter is cleared to 0000H in synchronization with the peripheral clock if the valid edge of the TECRm pin specified by the TTmECS1 and TTmECS0 bits is detected. At this time, an encoder clear interrupt request signal (INTTIECm) is generated. When the TTmSCE bit = 0, setting of the TTmZCL, TTmBCL, and TTmACL bits is invalid.

Figure 8-62. Operation Example (When TTmSCE Bit = 0, TTmECS1 and TTmECS0 Bits = 01, and TTmUDS1 and TTmUDS0 Bits = 11)


Encoder input (TENCm0 pin input)
Encoder input (TENCm1 pin input)
Encoder clear input (TECRm pin input)
Peripheral clock
TTmCNT register
N
N + 1
0000H
0001H
0002H
Count timing signal
INTTIECm
Counter clear

Remark m = 0, 1

(b) Clearing method <2>: By detecting clear level condition of the TENCm0, TENCm1, and TECRm pins (TTmSCE bit = 1)

When the TTmSCE bit = 1, the 16-bit counter is cleared to 0000H if the clear level condition of the TECRm, TENCm0, or TENCm1 pin specified by the TTmZCL, TTmBCL, and TTmACL bits is detected. At this time, the encoder clear interrupt request signal (INTTIECm) is not generated. Setting of the TTmECS1 and TTmECS0 bits is invalid when the TTmSCE bit = 1.

Table 8-11. 16-bit Counter Clearing Condition When TTmSCE Bit = 1

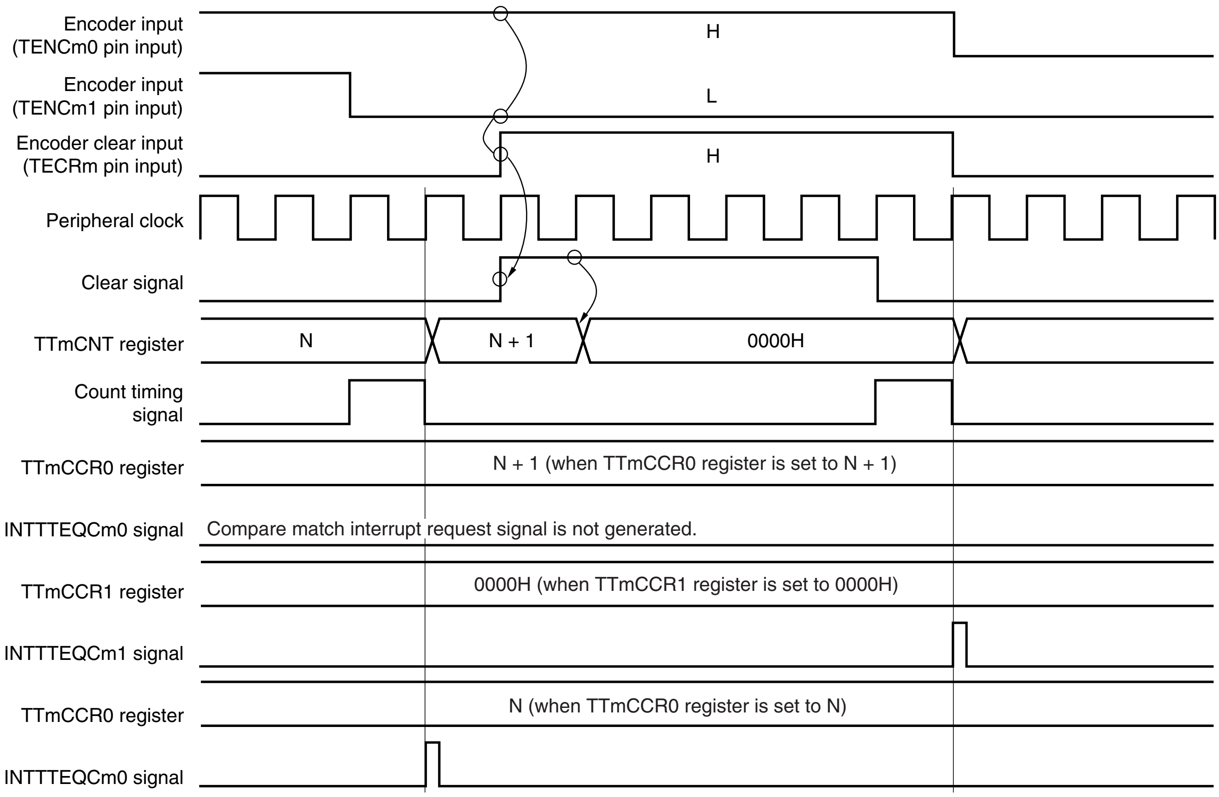
Clear Level Condition Setting			Input Level of Encoder Pin		
TTmZCL Bit	TTmBCL Bit	TTmACL Bit	TECRm Pin	TENCm1 Pin	TENCm0 Pin
0	0	0	L	L	L
0	0	1	L	L	H
0	1	0	L	H	L
0	1	1	L	H	H
1	0	0	H	L	L
1	0	1	H	L	H
1	1	0	H	H	L
1	1	1	H	H	H

Caution **The 16-bit counter is cleared to 0000H when the clear level condition of the TTmZCL, TTmBCL, and TTmACL bits match the input level of the TECRm, TENCm1, or TENCm0 pin.**

Remark m = 0, 1

Figure 8-63. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (1/3)

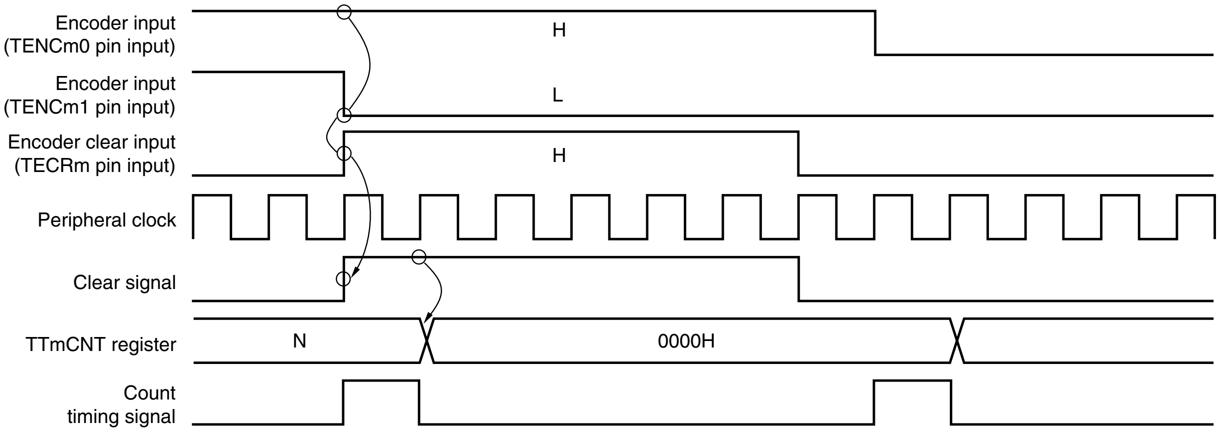
(i) If inputting the high level to the TECRm pin lags behind inputting the low level to the TENCm1 pin while the counter is counting up, the counter is cleared after it counts up.


Encoder input (TENCm0 pin input)
H
Encoder input (TENCm1 pin input)
L
Encoder clear input (TECRm pin input)
H
Peripheral clock
Clear signal
TTmCNT register
N
N + 1
0000H
Count timing signal
TTmCCR0 register
N + 1 (when TTmCCR0 register is set to N + 1)
INTTTEQCm0 signal
Compare match interrupt request signal is not generated.
TTmCCR1 register
0000H (when TTmCCR1 register is set to 0000H)
INTTTEQCm1 signal
TTmCCR0 register
N (when TTmCCR0 register is set to N)
INTTTEQCm0 signal

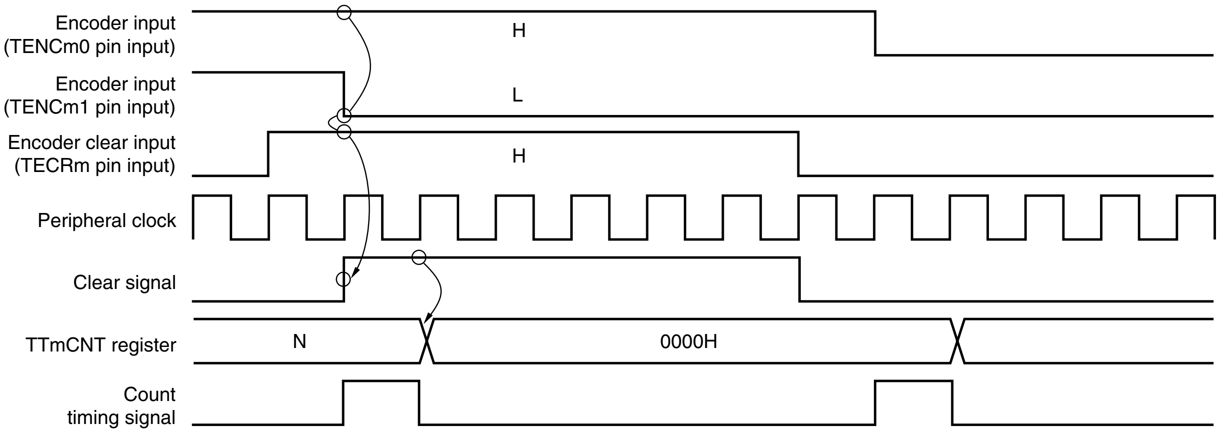
Remark m = 0, 1

Figure 8-63. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (2/3)

(ii) If the high level is input to the TECRm pin at the same time as the low level is input to the TECNm1 pin while the counter is counting up, the counter is cleared without counting up.


Encoder input (TENCm0 pin input)
H
Encoder input (TENCm1 pin input)
L
Encoder clear input (TECRm pin input)
H
Peripheral clock
Clear signal
TTmCNT register
N
0000H
Count timing signal

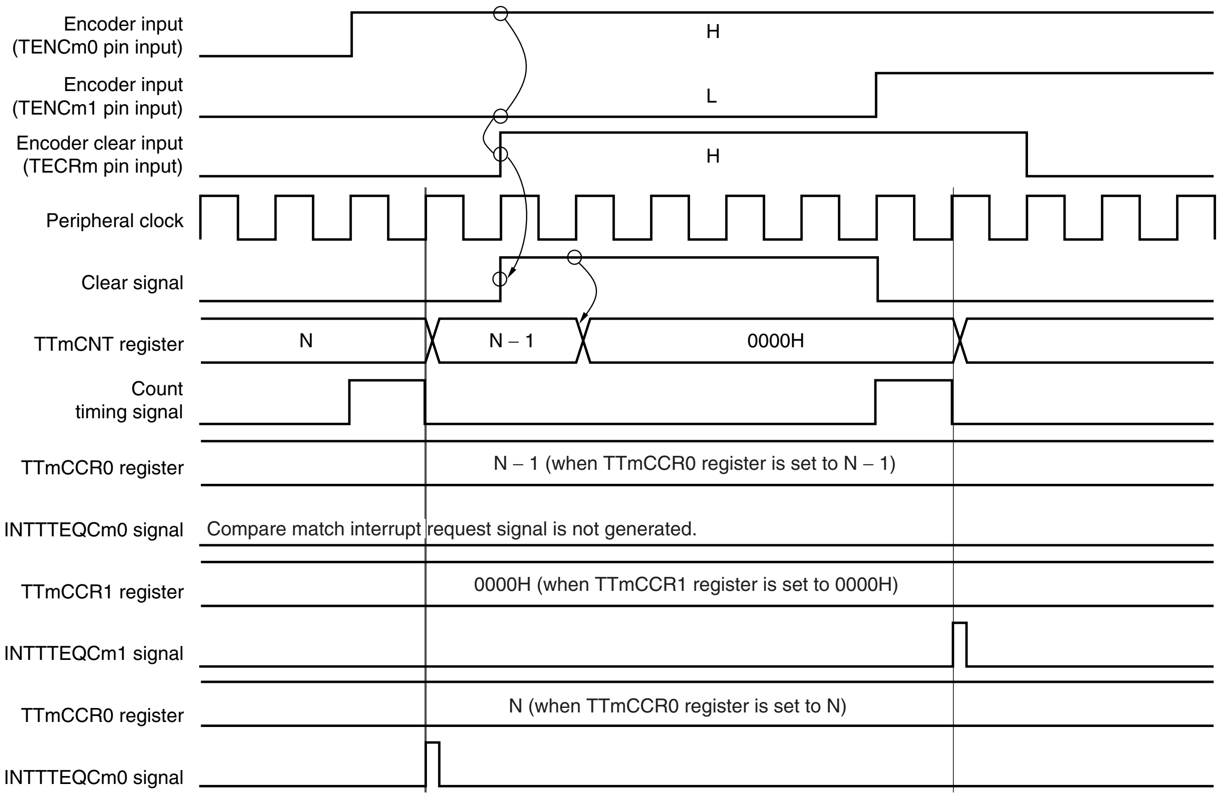
(iii) If the high level is input to the TECRm pin earlier than the low level is input to the TENCm1 pin while the counter is counting up, the counter is cleared without counting up.


Encoder input (TENCm0 pin input)
H
Encoder input (TENCm1 pin input)
L
Encoder clear input (TECRm pin input)
H
Peripheral clock
Clear signal
TTmCNT register
N
0000H
Count timing signal

Remark m = 0, 1

Figure 8-63. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (3/3)

(iv) If the high level is input to the TECRm pin later than the low level is input to the TENCm1 pin while the counter is counting up, the counter is cleared after it counts up.


Encoder input (TENCm0 pin input)
H
Encoder input (TENCm1 pin input)
L
Encoder clear input (TECRm pin input)
H
Peripheral clock
Clear signal
TTmCNT register
N
N – 1
0000H
Count timing signal
TTmCCR0 register
N – 1 (when TTmCCR0 register is set to N – 1)
INTTTEQCm0 signal
Compare match interrupt request signal is not generated.
TTmCCR1 register
0000H (when TTmCCR1 register is set to 0000H)
INTTTEQCm1 signal
TTmCCR0 register
N (when TTmCCR0 register is set to N)
INTTTEQCm0 signal

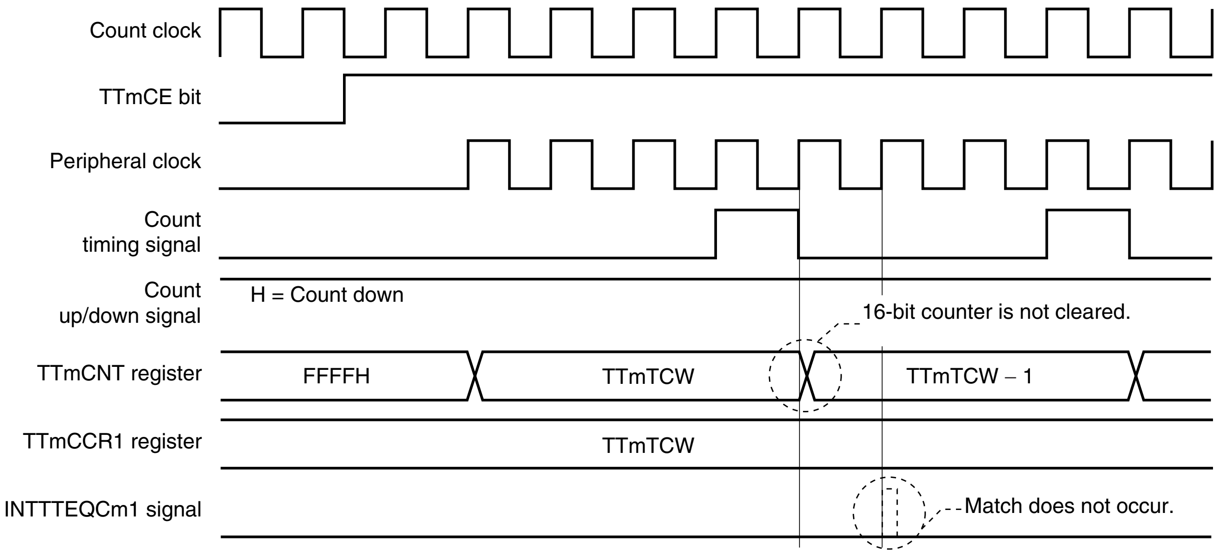
Remark m = 0, 1

If the counter is cleared in this way, a miscount does not occur even if inputting the signal to the TECRm pin is late, because the clear level condition of the TECRm, TENCm1, and TENCm0 pins is set and the 16-bit counter is cleared to 0000H when the clear level condition is detected.

(7) Notes on using encoder count function

(a) If compare match interrupt is not generated immediately after operation is started

If a value which is the same as that of the TTmTCW register is set to the TTmCCR0 or TTmCCR1 register and the counter operation is started when the TTmCTL2.TTmECC bit = 0, and if the count value (TTmTCW) of the 16-bit counter matches the value of the CCRa buffer register immediately after the start of the operation, the match is masked and the compare match interrupt request signal (INTTTEQCma) is not generated (a = 0, 1). In addition, the 16-bit counter is not cleared to 0000H by setting the TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits.

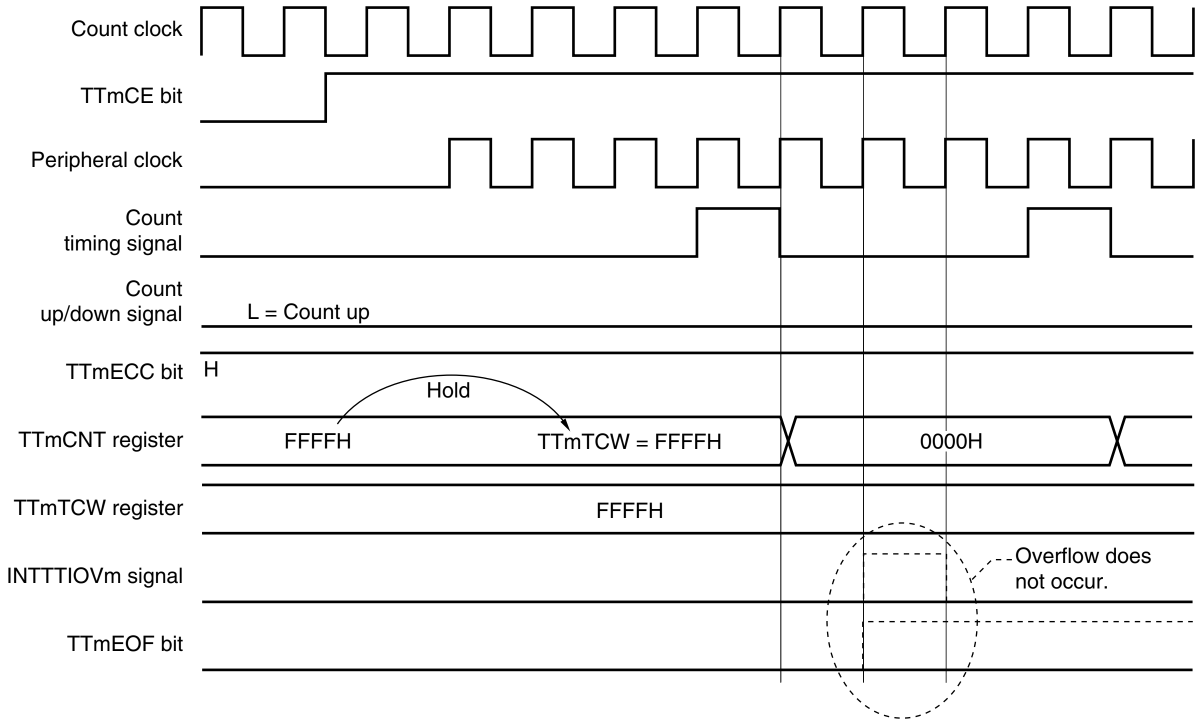


Remark m = 0, 1

(b) If overflow does not occur immediately after start of operation

If the count operation is resumed when the TTmCTL2.TTmECC bit = 1, the 16-bit counter does not overflow if its count value that has been held is FFFFH and if the next count operation is counting up.

After the counter starts operating and counts up from a count value (value of TTmTCW register = FFFFH), the counter overflows from FFFFH to 0000H. However, detection of the overflow is masked, the overflow flag (TTmEOF) is not set, and the overflow interrupt request signal (INTTTIOVm) is not generated.



Remark m = 0, 1

8.6.10 Encoder compare mode (TTmMD3 to TTmMD0 bits = 1000)

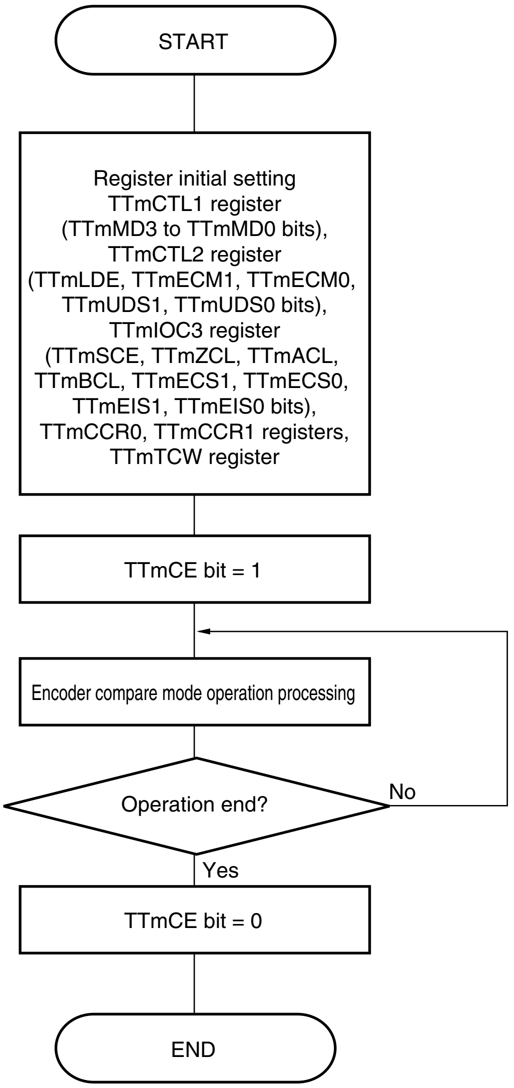
In the encoder compare mode, the encoder is controlled by using both the TTmCCR0 and TTmCCR1 registers as compare registers and the input pins for encoder count function (TENCm0, TENCm1, and TECRm).

In this mode, the 16-bit counter can be cleared to 0000H in three ways: when the count value of the counter matches the value of the CCRa buffer register (compare match interrupt request signal (INTTTEQCma) is generated), when the edge of the encoder clear input (TECRm pin) is detected and cleared, and when the clear level condition of TENCm0, TENCm1, and TECRm pins is detected and cleared.

When the 16-bit counter underflows, the set value of the TTmCCR0 register can be transferred to the counter.

(1) Encoder compare mode operation flow

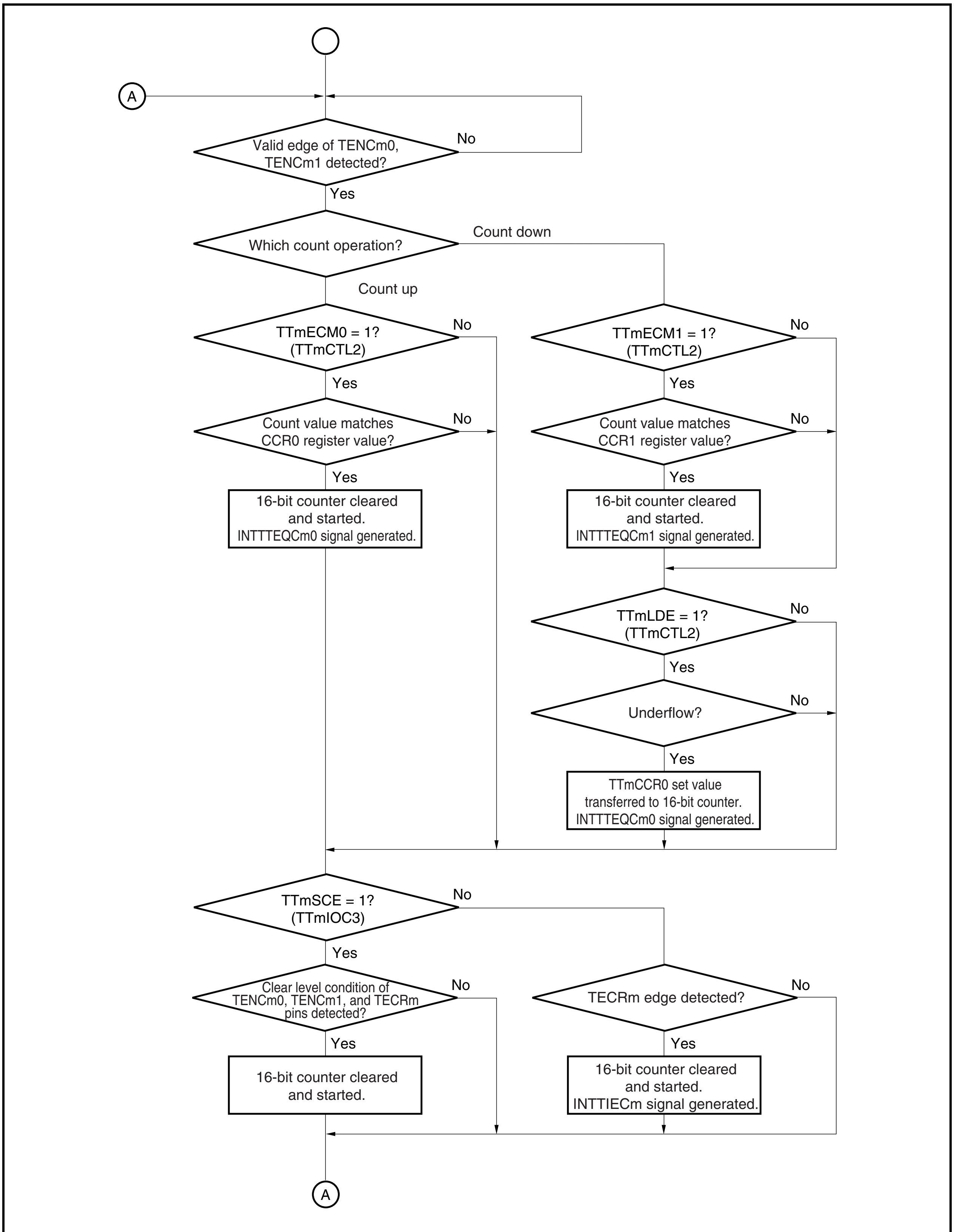
Figure 8-64. Encoder Compare Mode Operation Flow



: See **Figure 8-65 Encoder Compare Mode Operation Processing**.

Remark m = 0, 1

Figure 8-65. Encoder Compare Mode Operation Processing


A
Valid edge of TENCm0, TENCm1 detected?
No
Yes
Which count operation?
Count down
Count up
TTmECM0 = 1? (TTmCTL2)
No
Yes
Count value matches CCR0 register value?
No
Yes
16-bit counter cleared and started. INTTTEQCm0 signal generated.
TTmECM1 = 1? (TTmCTL2)
No
Yes
Count value matches CCR1 register value?
No
Yes
16-bit counter cleared and started. INTTTEQCm1 signal generated.
TTmLDE = 1? (TTmCTL2)
No
Yes
Underflow?
No
Yes
TTmCCR0 set value transferred to 16-bit counter. INTTTEQCm0 signal generated.
TTmSCE = 1? (TTmIOC3)
No
Yes
Clear level condition of TENCm0, TENCm1, and TECRm pins detected?
No
Yes
16-bit counter cleared and started.
TECRm edge detected?
No
Yes
16-bit counter cleared and started. INTTIECm signal generated.
A

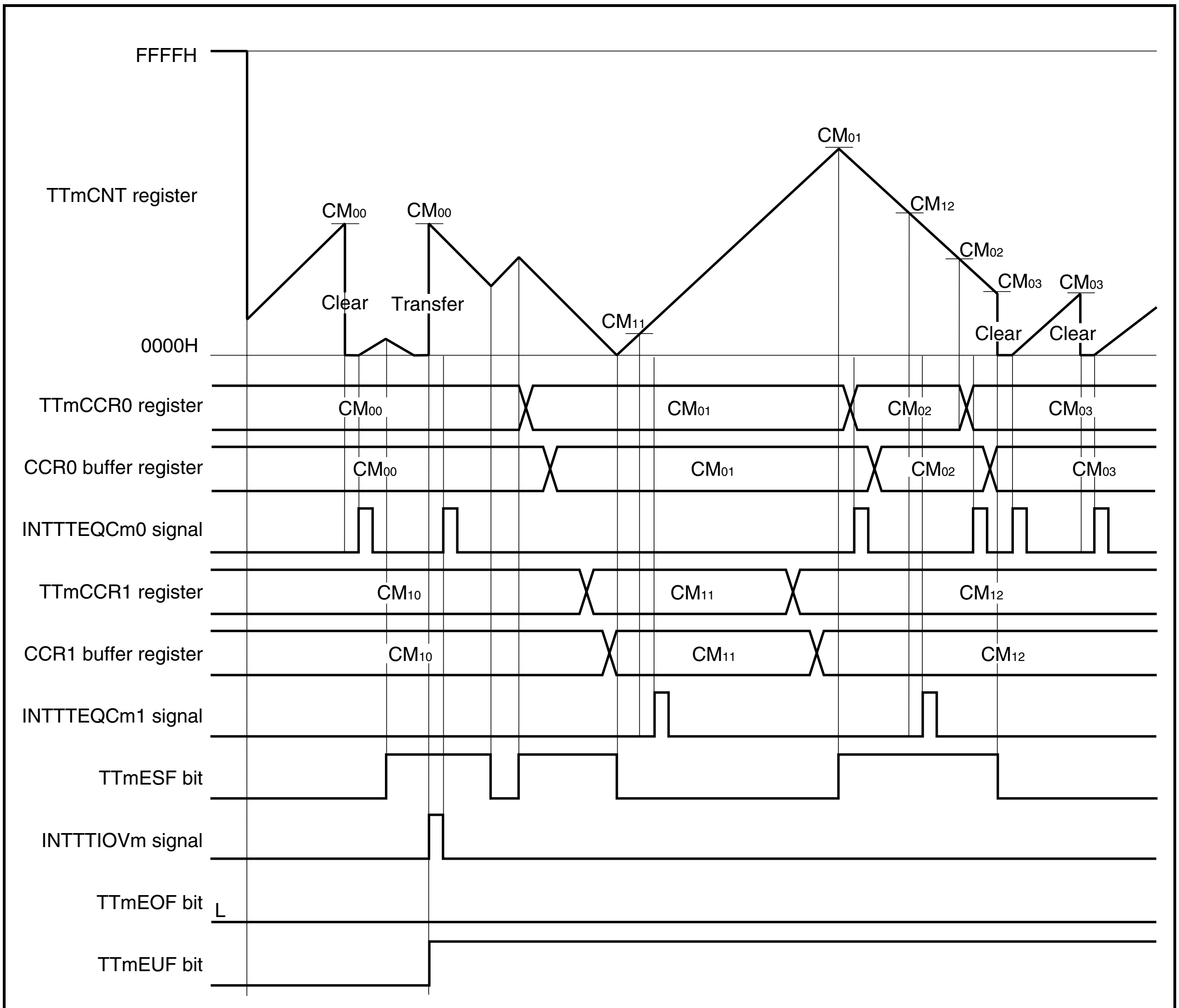
Remark m = 0, 1

(2) Encoder compare mode operation timing

(a) Basic timing 1

[Register setting conditions]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 01
  The 16-bit counter is cleared to 0000H when its count value matches the value of the CCR0 buffer register.
- TTmCTL2.TTmLDE bit = 1
  The set value of the TTmCCR0 register is transferred to the 16-bit counter when it overflows.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00
  Specification of the edge of encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



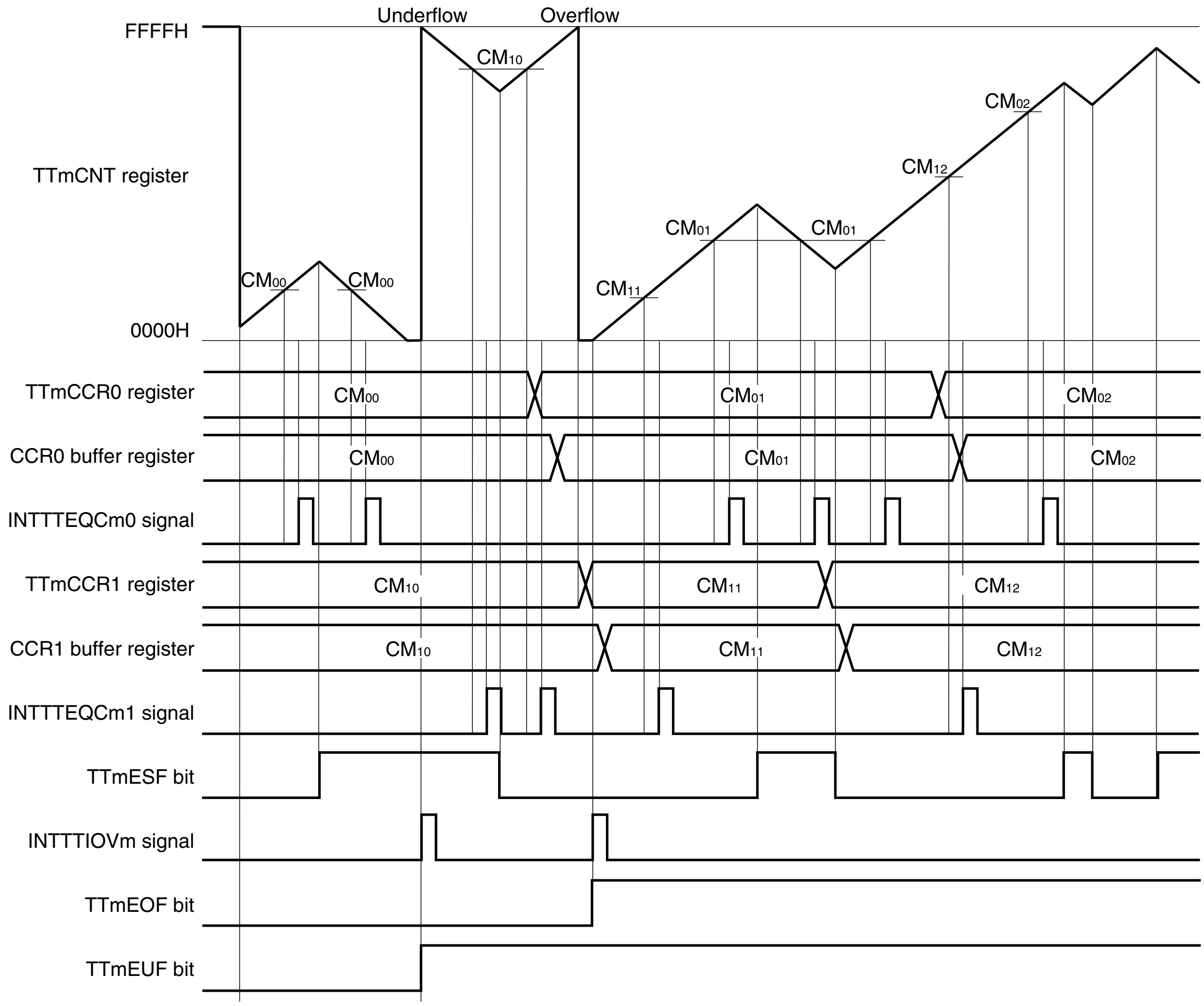
Remark m = 0, 1

When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the counter and the 16-bit counter starts operating.
When the count value of the counter matches the value of the CCR0 buffer register, the compare match interrupt request signal (INTTTEQCm0) is generated. Because the TTmECM0 bit = 1, the 16-bit counter is cleared to 0000H if the next count operation is counting up.
When the count value of the 16-bit counter matches the value of the CCR1 buffer register, the compare match interrupt request signal (INTTTEQCm1) is generated. Because the TTmECM1 bit = 0, the 16-bit counter is not cleared to 0000H when its value matches that of the CCR1 buffer register.
When the TTmLDE bit = 1 and TTmECM0 bit = 1, the counter can operate in a range from 0000H to the set value of the TTmCCR0 register.

(b) Basic timing 2

[Register setting condition]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 00
  The 16-bit counter is not cleared even when its count value matches the value of the CCRa buffer register (a = 0, 1).
- TTmCTL2.TTmLDE bit = 0
  The set value of the TTmCCR0 register is not transferred to the 16-bit counter after the counter underflows.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00
  Specification of the edge of the encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



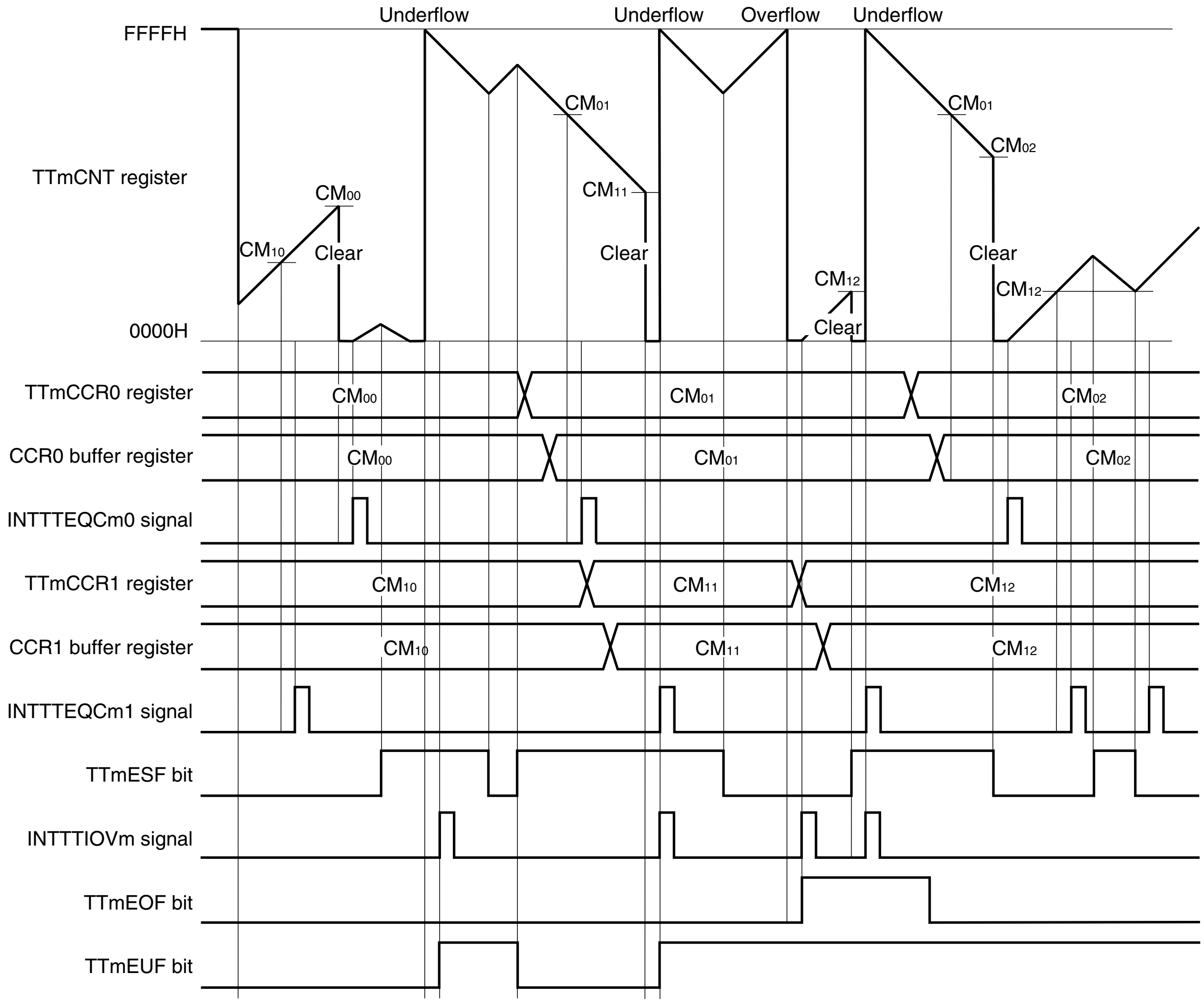
Remark m = 0, 1

When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the 16-bit counter and the counter starts operating.
When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCm0) is generated.
When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCm1) is generated.
The 16-bit counter is not cleared to 0000H even when its count value matches the value of the CCRa buffer register because the TTmECM1 and TTmECM0 bits = 00 (a = 0, 1).

(c) Basic timing 3

[Register setting condition]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 11
  The count value of the 16-bit counter is cleared to 0000H when its value matches the value of the CCR0 buffer register.
  The count value of the 16-bit counter is cleared to 0000H when its value matches the value of the CCR1 buffer register.
- Setting of the TTmCTL2.TTmLDE bit is invalid.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00
  Specification of the edge of the encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



Remark m = 0, 1

When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the 16-bit counter and the counter starts operating.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCm0) is generated. At this time, the 16-bit counter is cleared to 0000H if the next count operation is counting up.

When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCm1) is generated. At this time, the 16-bit counter is cleared to 0000H if the next count operation is counting down.

CHAPTER 9 16-BIT INTERVAL TIMER M (TMM)

Timer M (TMM) is a 16-bit interval timer.
The V850E/IG4-H and V850E/IH4-H incorporate TMM0 to TMM3.

9.1 Overview

An outline of TMMn is shown below (n = 0 to 3).

- Interval function
- 8 clocks selectable
- 16-bit counter × 1 (The 16-bit counter cannot be read during timer count operation.)
- Compare register × 1 (The compare register cannot be written during timer count operation.)
- Compare match interrupt × 1

Timer M supports only the clear & start mode. The free-running timer mode is not supported.

9.2 Configuration

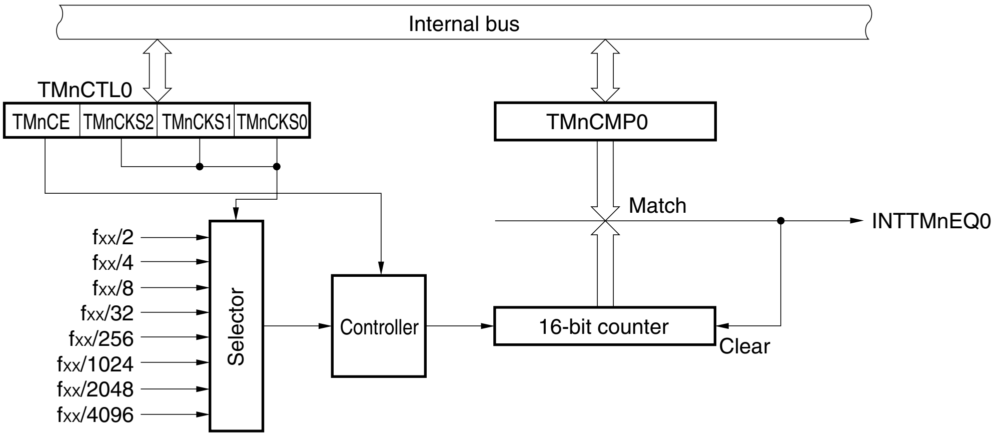
TMMn includes the following hardware (n = 0 to 3).

Table 9-1. Configuration of TMMn

Item	Configuration
Timer register	16-bit counter × 1
Register	TMMn compare register 0 (TMnCMP0)
Control register	TMMn control register 0 (TMnCTL0)

Remark n = 0 to 3

Figure 9-1. Block Diagram of TMMn



Remarks 1. fxx: Peripheral clock frequency

2. n = 0 to 3

(1) 16-bit counter

This is a 16-bit counter that counts the internal clock.

The 16-bit counter cannot be read or written.

(2) TMMn compare register 0 (TMnCMP0)

The TMnCMP0 register is a 16-bit compare register.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

The same value can always be written to the TMnCMP0 register by software.

Rewriting the TMnCMP0 register is prohibited during TMMn operation (TMnCTL0.TMnCE bit = 1).

After reset: 0000H R/W Address: TM0CMP0 FFFFF544H, TM1CMP0 FFFFF554H,
TM2CMP0 FFFFF564H, TM3CMP0 FFFFF574H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMnCMP0 (n = 0 to 3)																

9.3 Control Register

(1) TMMn control register 0 (TMnCTL0)

The TMnCTL0 register is an 8-bit register that controls the TMMn operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

The same value can always be written to the TMnCTL0 register by software.

After reset: 00H R/W Address: TM0CTL0 FFFFF540H, TM1CTL0 FFFFF550H,
TM2CTL0 FFFFF560H, TM3CTL0 FFFFF570H

	<7>	6	5	4	3	2	1	0
TMnCTL0 (n = 0 to 3)	TMnCE	0	0	0	0	TMnCKS2	TMnCKS1	TMnCKS0

TMnCE	Internal clock operation enable/disable specification
0	TMMn operation disabled (16-bit counter reset asynchronously)
1	TMMn operation enabled. Start operation clock supply. Start TMMn operation.
The internal clock control and internal circuit reset for TMMn are performed asynchronously with the TMnCE bit. When the TMnCE bit is cleared to 0, the internal clock of TMMn is stopped (fixed to low level) and 16-bit counter is reset asynchronously.	

TMnCKS2	TMnCKS1	TMnCKS0	Count clock selection
0	0	0	fxx/2
0	0	1	fxx/4
0	1	0	fxx/8
0	1	1	fxx/32
1	0	0	fxx/256
1	0	1	fxx/1024
1	1	0	fxx/2048
1	1	1	fxx/4096

Cautions 1. Set the TMnCKS2 to TMnCKS0 bits when the TMnCE bit = 0.
However, when changing the value of the TMnCE bit from 0 to 1, it is impossible to set the value of the TMnCKS2 to TMnCKS0 bits simultaneously.

2. Be sure to clear bits 3 to 6 to “0”.

Remark fxx: Peripheral clock frequency

9.4 Operation

9.4.1 Interval timer mode

In the interval timer mode, an interrupt request signal (INTTMnEQ0) is generated at the interval set by the TMnCMP0 register if the TMnCTL0.TMnCE bit is set to 1.

Figure 9-2. Configuration of Interval Timer

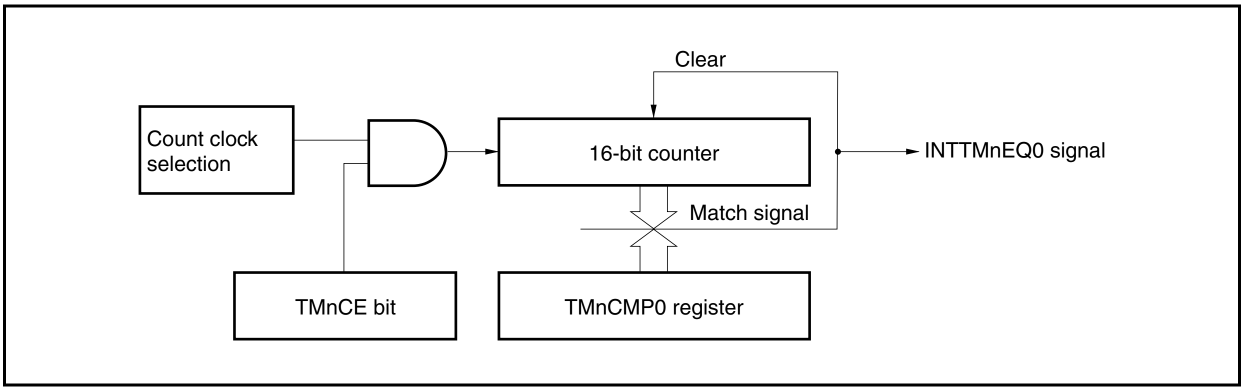
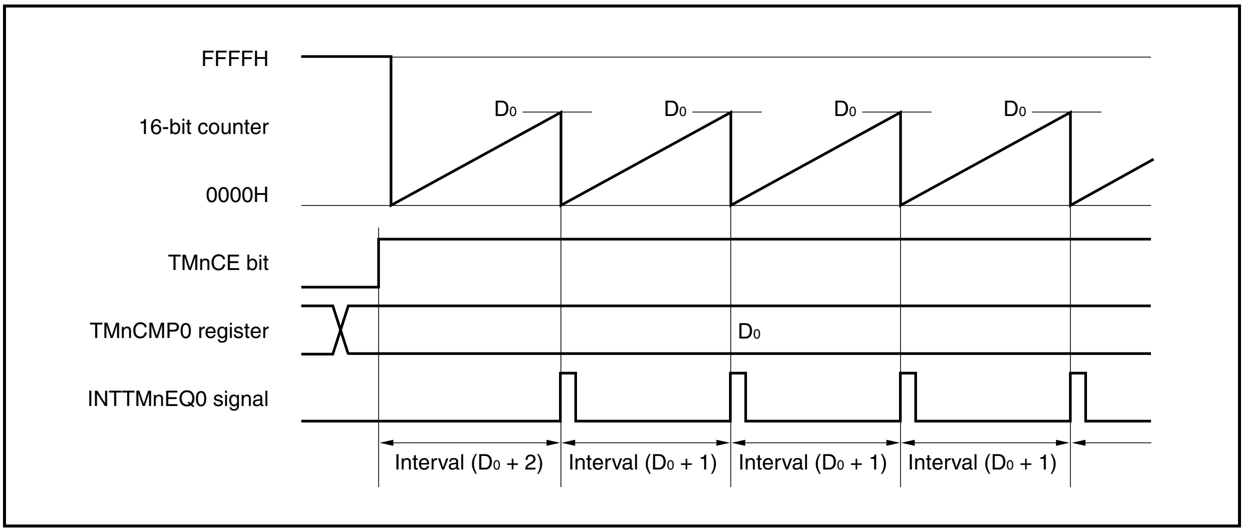


Figure 9-3. Basic Timing of Operation in Interval Timer Mode



When the TMnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting.

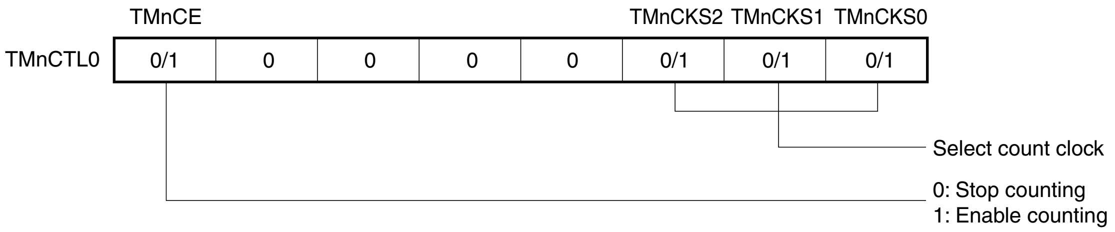
When the count value of the 16-bit counter matches the value of the TMnCMP0 register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTMnEQ0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TMnCMP0 register + 1) × Count clock cycle

Figure 9-4. Register Setting for Interval Timer Mode Operation

(a) TMMn control register 0 (TMnCTL0)



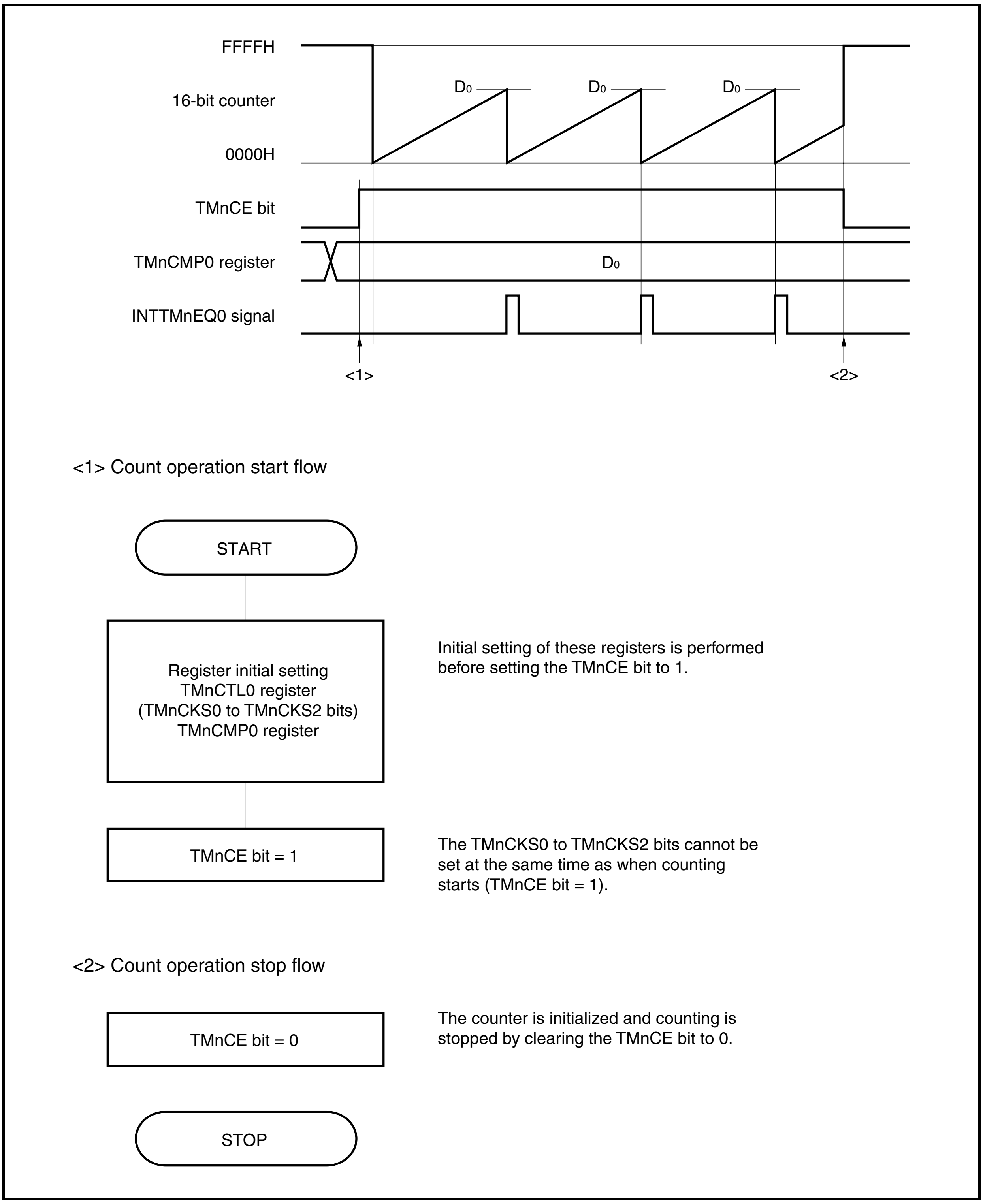
(b) TMMn compare register 0 (TMnCMP0)

If the TMnCMP0 register is set to D_0, the interval is as follows.

Interval = (D_0 + 1) × Count clock cycle

(1) Interval timer mode operation flow

Figure 9-5. Software Processing Flow in Interval Timer Mode



<1> Count operation start flow

Initial setting of these registers is performed before setting the TMnCE bit to 1.

The TMnCKS0 to TMnCKS2 bits cannot be set at the same time as when counting starts (TMnCE bit = 1).

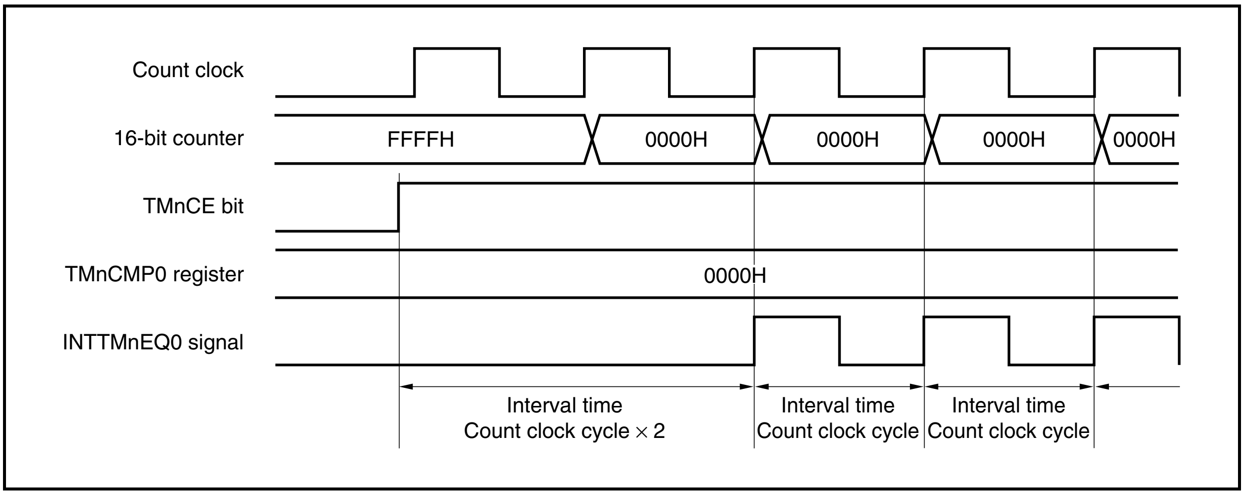
<2> Count operation stop flow

The counter is initialized and counting is stopped by clearing the TMnCE bit to 0.

(2) Interval timer mode operation timing

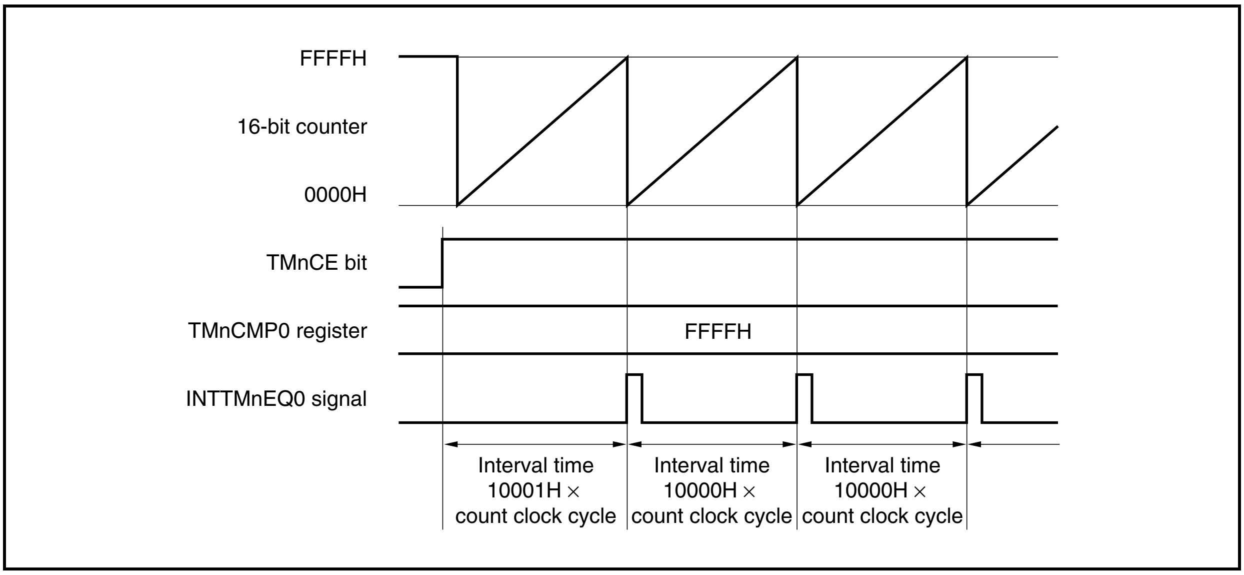
(a) Operation if TMnCMP0 register is set to 0000H

If the TMnCMP0 register is set to 0000H, the INTTMnEQ0 signal is generated at each count clock. The value of the 16-bit counter is always 0000H.



(b) Operation if TMnCMP0 register is set to FFFFH

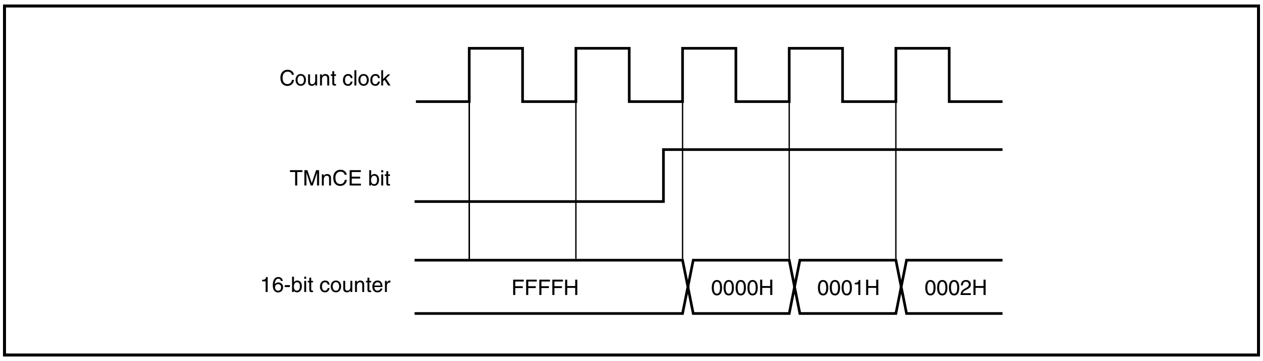
If the TMnCMP0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTMnEQ0 signal is generated.



9.5 Cautions

(1) Error on starting timer

It takes one clock to generate the first compare match interrupt request signal (INTTMnEQ0) after the TMnCTL0.TMnCE bit is set to 1 and TMMn is started. This is because the value of the 16-bit counter is FFFFH when the TMnCE bit = 0 and TMMn is started asynchronously to the count clock.



(2) Rewriting the TMnCMP0 and TMnCTL0 registers is prohibited while TMMn is operating.

If these registers are rewritten while the TMnCTL0.TMnCE bit is 1, the operation cannot be guaranteed. If they are rewritten by mistake, clear the TMnCE bit to 0, and re-set the registers.

CHAPTER 10 MOTOR CONTROL FUNCTION

10.1 Functional Overview

Timer ABn (TABn) and the TMQn option (TMQOPn) can be used as an inverter function that controls a motor. It performs a tuning operation with timer AAn (TAAn) and A/D conversion of A/D converters 0 and 1 can be started when the value of TABn matches the value of TAAn. The following operations can be performed as motor control functions.

- 6-phase PWM output function with 16-bit accuracy (with dead-timer, for upper and lower arms)
- Timer tuning operation function (tunable with TAAn)
- Period setting function (period can be changed during operation of crest or valley interrupt)
- Compare register rewriting: Anytime rewrite, batch write, or intermittent rewrite (selectable during TABn operation)
- Interrupt and transfer culling functions
- Dead-time setting function
- A/D trigger timing function of A/D converters 0 and 1 (four types of timing can be generated)
- 0% output and 100% output available
- 0% output and 100% output selectable by crest interrupt and valley interrupt
- Forced output stop function
    - At valid edge detection by external pin input (TOBnOFF, TOB0OFF, TOTmOFF)
    - At overvoltage detection by comparator function of A/D converter
    - At main clock oscillation stop detection by clock monitor function

Remark V850E/IG4-H: n = 0, m = 2, 3
V850E/IH4-H: n = 0, 1, m = 2, 3

10.2 Configuration

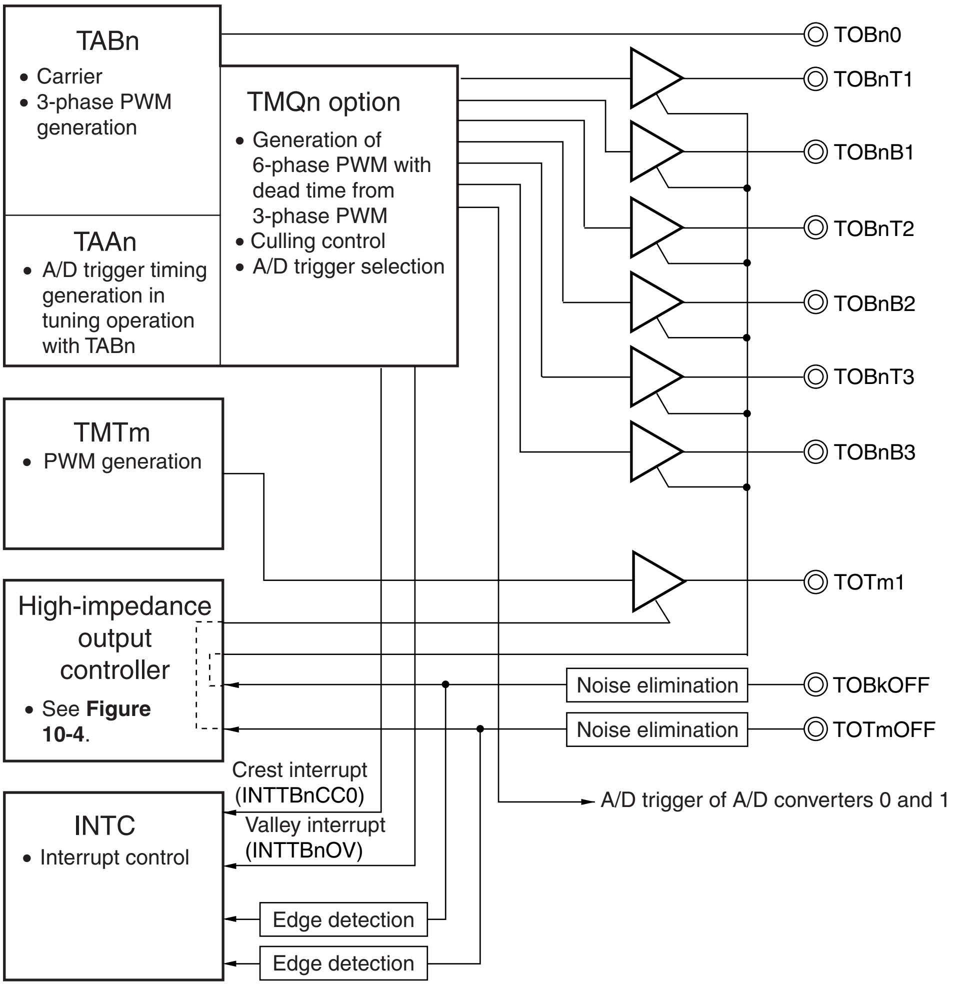
The motor control function consists of the following hardware.

Item	Configuration
Timer register	Dead-time counter m
Compare register	TABn dead-time compare register (TABnDTC register)
Control registers	TABn option register 0 (TABnOPT0) TABn option register 1 (TABnOPT1) TABn option register 2 (TABnOPT2) TABn option register 3 (TABnOPT3) TABn I/O control register 3 (TABnIOC3) High-impedance output control registers 0, 1 (HZAyCTLa)

Remark V850E/IG4-H: m = 0 to 3, n = 0, y = 0 to 12, a = 0, 1
V850E/IH4-H: m = 0 to 3, n = 0, 1, y = 0 to 12, a = 0, 1

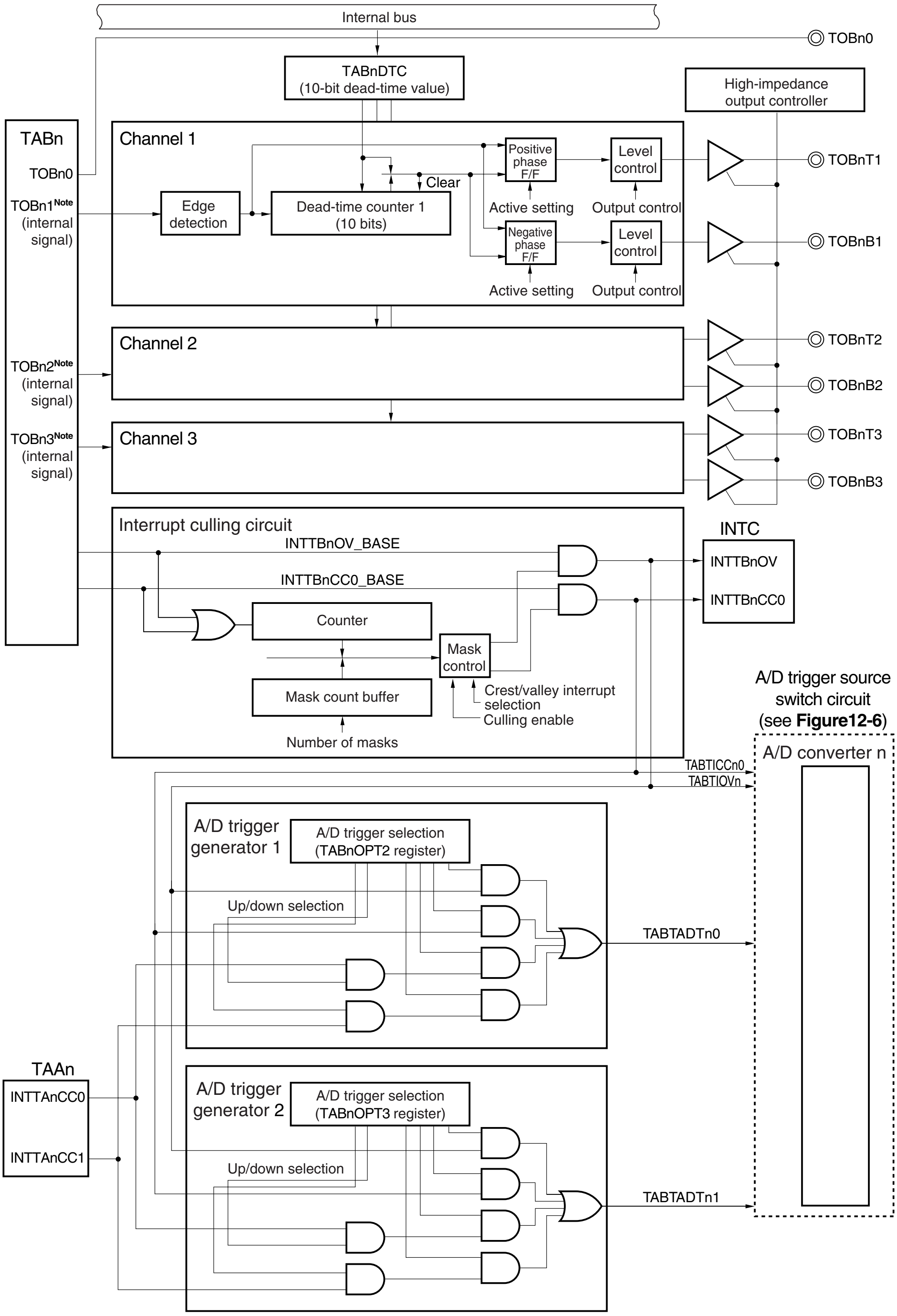
- 6-phase PWM output can be produced with dead time by using the output of TABn (TOBn1, TOBn2, TOBn3)
- The output level of the 6-phase PWM output can be set individually.
- The 16-bit timer/counter of TABn counts up/down triangular waves. When the timer/counter underflows and when a period match occurs, an interrupt is generated. Interrupt generation, however, can be culled up to 31 times.
- TAAn can execute counting at the same time as TABn (timer tuning operation function). TAAn can be set in four ways as it can generate two types of A/D trigger sources (INTTAnCC0 and INTTAnCC1), and two types of interrupts: on underflow interrupt of TABn (INTTBnOV) and period match interrupt (INTTBnCC0).

Figure 10-1. Block Diagram of Motor Control



Remark V850E/IG4-H: n = 0, m = 2, 3, k = 0, 1, 01
V850E/IH4-H: n = 0, 1, m = 2, 3, k = 0, 1, 01

Figure 10-2. TMQn Option



Note TOBn1, TOBn2, and TOBn3 function alternately as output pins.

Remark V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

(1) TABn dead-time compare register (TABnDTC)

The TABnDTC register is a 10-bit compare register that specifies a dead-time value.

Rewriting this register is prohibited when the TABnCTL0.TABnCE bit = 1.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Caution To generate a dead time period, set a value of 1 or greater to the TABnDTC register.
While the operation is stopped (TABnCTL0.TABnCE bit = 0), the dead time period is not generated and the output levels of the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins are in the initial status. To protect the system, therefore, allow the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins to go into a high-impedance state or select the port mode with setting the output levels of the pins, before stopping the operation.
If the dead time period is not necessary, set the TABnDTC register to 0.

After reset: 0000H R/W Address: TAB0DTC FFFFF604H, TAB1DTC FFFFF644H[Note]

	15 … 10	9 … 0
TABnDTC	000000	TABnDTC9 to TABnDTC0

(V850E/IG4-H
n = 0)

(V850E/IH4-H
n = 0, 1)

Note V850E/IH4-H only

(2) Dead-time counters 1 to 3

The dead-time counters are 10-bit counters that count dead time.

These counters are cleared or count up at the rising or falling edge of the TOBnm output signal by TABn, and are cleared and stopped when their count value matches the value of the TABnDTC register. The count clock of these counters is the same as that set by the TABnCTL0.TABnCKS2 to TABnCTL0.TABnCKS0 bits of TABn.

Remarks 1. The operation differs when the TABnOPT2.TABnDTM bit = 1. For details, see **10.4.2 (4) Automatic dead-time width narrowing function (TABnOPT2.TABnDTM bit = 1)**.

2. V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

10.3 Control Registers

(1) TABn option register 0 (TABnOPT0)

The TABnOPT0 register is an 8-bit register that controls the timer Qn option function.

This register can be read or written in 8-bit or 1-bit units. However, the TABnCUF bit is read-only.

Reset sets this register to 00H.

Caution The TABnCMS and TABnCUF bits can be set only in the 6-phase PWM output mode. Be sure to clear these bits to 0 when TABn is used alone (V850E/IG4-H: n = 0, V850E/IH4-H: n = 0, 1)

After reset: 00H R/W Address: TAB0OPT0 FFFFF5E5H, TAB1OPT0 FFFFF625H

	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TABnOPT0	TABmCCS3[Notes 1, 2]	TABmCCS2[Notes 1, 2]	TABmCCS1[Notes 1, 2]	TABmCCS0[Notes 1, 2]	0	TABnCMS[Note 3]	TABnCUF[Note 3]	TABnOVF[Note 4]

(V850E/IG4-H n = 0, 1 m = 0)

(V850E/IH4-H n = 0, 1 m = 0, 1)

TABnCMS[Note 3]	Compare register rewrite mode selection
0	Batch write mode (transfer operation)
1	Anytime write mode

- The TABnCMS bit is valid only when the 6-phase PWM output mode is set (when the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 111). Clear the TABnCMS bit to 0 in any other mode.
- The TABnCMS bit can be rewritten while the timer is operating (when the TABnCTL0.TABnCE bit = 1).
- The following compare registers are rewritten in the batch write mode. TABnCCR0 to TABnCCR3, TAnCCR0, TAnCCR1, TABnOPT1, and TABnDTC registers

TABnCUF[Note 3]	Up-count/down-count flag of timer ABn
0	Timer ABn is counting up.
1	Timer ABn is counting down.

The TABnCUF bit is valid only when the 6-phase PWM output mode is set (when the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 111).

Notes 1. In the V850E/IG4-H, only TAB0 can be set.
Be sure to set bits 4 to 7 of TAB1 to 0.

2. Be sure to clear the TABmCCS3 to TABmCCS0 bits to 0 in the 6-phase PWM output mode.

3. In the V850E/IG4-H, be sure to set bits 1 and 2 of TAB1 to 0.

4. For details of the TABnOVF bit, see **CHAPTER 7 16-BIT TIMER/EVENT COUNTER AB (TAB)**.

(2) TABn option register 1 (TABnOPT1)

The TABnOPT1 register is an 8-bit register that controls the interrupt request signal generated by the timer Qn option function.

The TABnOPT1 register generates the signals output to the interrupt culling circuit, A/D trigger generator 1, and A/D trigger generator 2 shown in Figure 10-2.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1.

Two rewriting modes (batch write mode and anytime write mode) can be selected, depending on the setting of the TABnOPT0.TABnCMS bit.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0OPT1 FFFFF600H, TAB1OPT1 FFFFF640H[Note 1]

	<7>	<6>	5	4	3	2	1	0
TABnOPT1	TABnICE	TABnIOE	0	TABnID4	TABnID3	TABnID2	TABnID1	TABnID0

(V850E/IG4-H n = 0)

(V850E/IH4-H n = 0, 1)

TABnICE	Crest interrupt (INTTBnCC0 signal) enable[Note 2]
0	Do not use INTTBnCC0 signal (do not use it as count signal for interrupt culling).
1	Use INTTBnCC0 signal (use it as count signal for interrupt culling).

TABnIOE	Valley interrupt (INTTBnOV signal) enable[Note 2]
0	Do not use INTTBnOV signal (do not use it as count signal for interrupt culling).
1	Use INTTBnOV signal (use it as count signal for interrupt culling).

TABnID4	TABnID3	TABnID2	TABnID1	TABnID0	Number of times of interrupt
0	0	0	0	0	Not culled (all interrupts are output)
0	0	0	0	1	1 masked (one of two interrupts is output)
0	0	0	1	0	2 masked (one of three interrupts is output)
0	0	0	1	1	3 masked (one of four interrupts is output)
:	:	:	:	:	:
1	1	1	0	0	28 masked (one of 29 interrupts is output)
1	1	1	0	1	29 masked (one of 30 interrupts is output)
1	1	1	1	0	30 masked (one of 31 interrupts is output)
1	1	1	1	1	31 masked (one of 32 interrupts is output)

Notes 1. V850E/IH4-H only

2. When using the crest interrupt (INTTBnCC0 signal) and the valley interrupt (INTTBnOV signal) as the count signal for interrupt culling or as the A/D trigger signal, set the signal to be used to 1.
An A/D trigger is generated at the culled interrupt timing.

(3) TABn option register 2 (TABnOPT2)

The TABnOPT2 register is an 8-bit register that controls the timer Qn option function.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1. However, rewriting the TABnDTM bit is prohibited when the TABnCE bit is 1. The same value can be rewritten.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAB0OPT2 FFFFF601H, TAB1OPT2 FFFFF641H[Note]

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TABnOPT2	TABnRDE	TABnDTM	TABnATM3	TABnATM2	TABnAT3	TABnAT2	TABnAT1	TABnAT0

V850E/IG4-H
n = 0
m = 1 to 3

V850E/IH4-H
n = 0, 1
m = 1 to 3

TABnRDE	Transfer culling enable
0	Do not cull transfer (transfer timing is generated every time at crest and valley).
1	Cull transfer at the same interval as interrupt culling set by the TABnOPT1 register.

TABnDTM	Dead-time counter operation mode selection
0	Dead-time counter counts up normally and, if TOBnm output of TABn is at a narrow interval (TOBnm output width < dead-time width), the dead-time counter is cleared and counts up again.
1	Dead-time counter counts up normally and, if TOBnm output of TABn is at a narrow interval (TOBnm output width < dead-time width), the dead-time counter counts down and the dead-time control width is automatically narrowed.
Rewriting the TABnDTM bit is disabled during timer operation. If it is rewritten by mistake, stop the timer operation by clearing the TABnCE bit to 0, and re-set the TABnDTM bit.	

Note V850E/IH4-H only

Cautions 1. When using interrupt culling (the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits are set to other than 00000), be sure to set the TABnRDE bit to 1.

Therefore, the interrupt and transfer are generated at the same timing. The interrupt and transfer cannot be set separately. If the interrupt and transfer are set separately (TABnRDE bit = 0), transfer is not performed normally.

2. To generate a dead time period, set a value 1 or greater to the TABnDTC register.

While the operation is stopped (TABnCTL0.TABnCE bit = 0), the dead time period is not generated and the output levels of the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins are in the initial status. To protect the system, therefore, allow the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins go into a high-impedance state or select the port mode with setting the output levels of the pins, before stopping the operation.

If the dead time period is not necessary, set the TABnDTC register to 0.

(2/2)

TABnATM3	TABnATM3 mode selection
0	Output A/D trigger signal (TABTADTn0) for INTTAnCC1 interrupt while 16-bit counter is counting up.
1	Output A/D trigger signal (TABTADTn0) for INTTAnCC1 interrupt while 16-bit counter is counting down.

TABnATM2	TABnATM2 mode selection
0	Output A/D trigger signal (TABTADTn0) for INTTAnCC0 interrupt while 16-bit counter is counting up.
1	Output A/D trigger signal (TABTADTn0) for INTTAnCC0 interrupt while 16-bit counter is counting down.

TABnAT3^Note^	A/D trigger output control 3
0	Disable output of A/D trigger signal (TABTADTn0) for INTTAnCC1 interrupt.
1	Enable output of A/D trigger signal (TABTADTn0) for INTTAnCC1 interrupt.

TABnAT2^Note^	A/D trigger output control 2
0	Disable output of A/D trigger signal (TABTADTn0) for INTTAnCC0 interrupt.
1	Enable output of A/D trigger signal (TABTADTn0) for INTTAnCC0 interrupt.

TABnAT1^Note^	A/D trigger output control 1
0	Disable output of A/D trigger signal (TABTADTn0) for INTTBnCC0 (crest interrupt).
1	Enable output of A/D trigger signal (TABTADTn0) for INTTBnCC0 (crest interrupt).

TABnAT0^Note^	A/D trigger output control 0
0	Disable output of A/D trigger signal (TABTADTn0) for INTTBnOV (valley interrupt).
1	Enable output of A/D trigger signal (TABTADTn0) for INTTBnOV (valley interrupt).

Note For the setting of the TABnAT3 to TABnAT0 bits, see **CHAPTER 12 A/D CONVERTERS 0 AND 1**.

(4) TABn option register 3 (TABnOPT3)

The TABnOPT3 register is an 8-bit register that controls the timer Qn option function.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0OPT3 FFFFF603H, TAB1OPT3 FFFFF643H[Note 1]

	7	6	<5>	<4>	<3>	<2>	<1>	<0>
TABnOPT3	0	0	TABnATM7	TABnATM6	TABnAT7	TABnAT6	TABnAT5	TABnAT4

(V850E/IG4-H n = 0)

(V850E/IH4-H n = 0, 1)

TABnATM7	TABnATM7 mode selection
0	Output A/D trigger signal (TABTADTn1) of INTTAnCC1 interrupt while 16-bit counter is counting up.
1	Output A/D trigger signal (TABTADTn1) of INTTAnCC1 interrupt while 16-bit counter is counting down.

TABnATM6	TABnATM6 mode selection
0	Output A/D trigger signal (TABTADTn1) of INTTAnCC0 interrupt while 16-bit counter is counting up.
1	Output A/D trigger signal (TABTADTn1) of INTTAnCC0 interrupt while 16-bit counter is counting down.

TABnAT7[Note 2]	A/D trigger output control 3
0	Disable output of A/D trigger signal (TABTADTn1) for INTTAnCC1 interrupt.
1	Enable output of A/D trigger signal (TABTADTn1) for INTTAnCC1 interrupt.

TABnAT6[Note 2]	A/D trigger output control 2
0	Disable output of A/D trigger signal (TABTADTn1) for INTTAnCC0 interrupt.
1	Enable output of A/D trigger signal (TABTADTn1) for INTTAnCC0 interrupt.

TABnAT5[Note 2]	A/D trigger output control 1
0	Disable output of A/D trigger signal (TABTADTn1) for INTTBnCC0 interrupt (crest interrupt).
1	Enable output of A/D trigger signal (TABTADTn1) for INTTBnCC0 interrupt (crest interrupt).

TABnAT4[Note 2]	A/D trigger output control 0
0	Disable output of A/D trigger signal (TABTADTn1) for INTTBnOV interrupt (valley interrupt).
1	Enable output of A/D trigger signal (TABTADTn1) for INTTBnOV interrupt (valley interrupt).

Notes 1. V850E/IH4-H only

2. For the setting of the TABnAT7 to TABnAT4 bits, see **CHAPTER 12 A/D CONVERTERS 0 AND 1**.

(5) TABn I/O control register 3 (TABnIOC3)

The TABnIOC3 register is an 8-bit register that controls the output of the timer Qn option function.

To output from the TOBnTm pin, set the TABnIOC0.TABnOEm bit to 1 and then set the TABnIOC3 register.

The TABnIOC3 register can be rewritten only when the TABnCTL0.TABnCE bit is 0.

Rewriting each bit of the TABnIOC3 register is prohibited when the TABnCTL0.TABnCE bit is 1; however the same value can be rewritten to each bit of the TABnIOC3 register when the TABnCTL0.TABnCE bit is 1.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to A8H.

Caution Set the TABnIOC3 register to the default value (A8H) when the timer is used in a mode other than the 6-phase PWM output mode.

Remark Set the output level of the TOBnTm pin by the TABnIOC0 register.

After reset: A8H R/W Address: TAB0IOC3 FFFFF602H, TAB1IOC3 FFFFF642H[Note]

	<7>	<6>	<5>	<4>	<3>	<2>	1	0
TABnIOC3	TABnOLB3	TABnOEB3	TABnOLB2	TABnOEB2	TABnOLB1	TABnOEB1	0	0

V850E/IG4-H
n = 0
m = 1 to 3

V850E/IH4-H
n = 0, 1
m = 1 to 3

TABnOLBm	Setting of TOBnBm pin output level
0	Disable inversion of output of TOBnBm pin
1	Enable inversion of output of TOBnBm pin

TABnOEBm	Setting of TOBnBm pin output
0	Disable TOBnBm pin output. • When TABnOLBm bit = 0, low level is output from TOBnBm pin. • When TABnOLBm bit = 1, high level is output from TOBnBm pin.
1	Enable TOBnBm pin output.

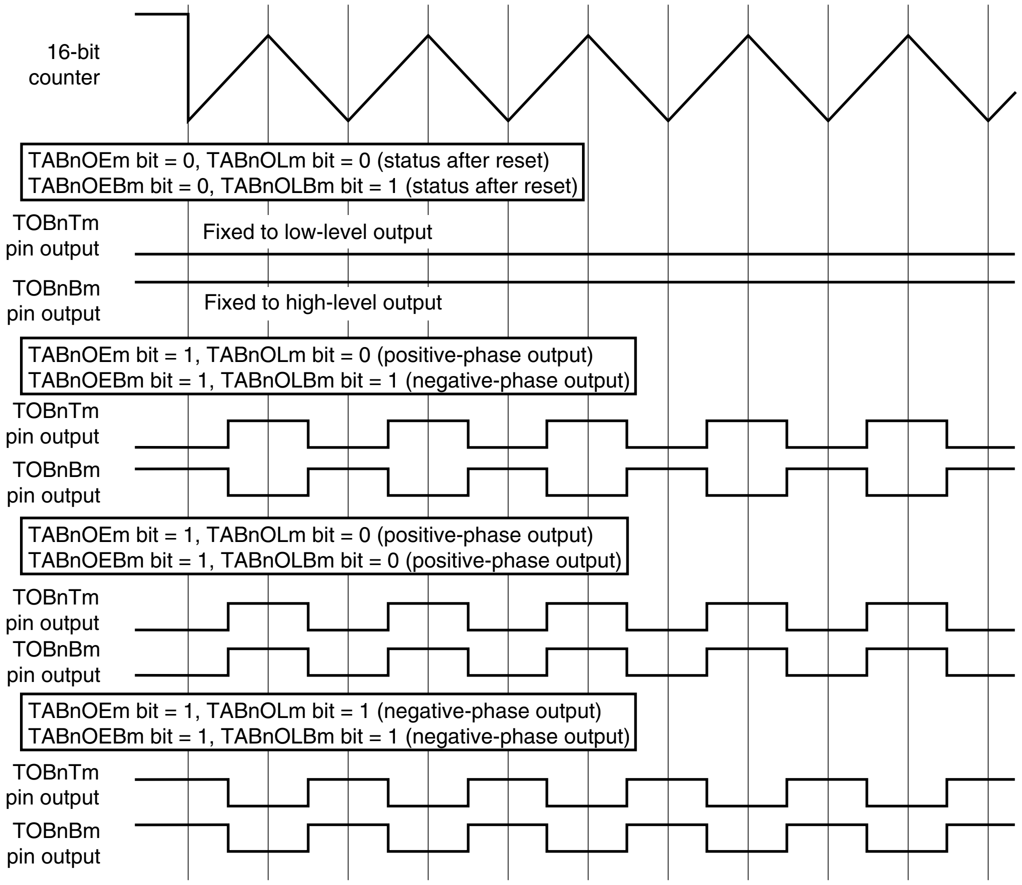
Note V850E/IH4-H only

(a) Output from TOBnTm and TOBnBm pins

The TOBnTm pin output is controlled by the TABnIOC0.TABnOLm and TABnIOC0.TABnOEm bits. The TOBnBm pin output is controlled by the TABnIOC3.TABnOLBm and TABnIOC3.TABnOEBm bits.

A timer output with each setting in the 6-phase PWM output mode is shown below.

Figure 10-3. TOBnTm and TOBnBm Pin Output Control (Without Dead Time)


16-bit counter
TABnOEm bit = 0, TABnOLm bit = 0 (status after reset)
TABnOEBm bit = 0, TABnOLBm bit = 1 (status after reset)
TOBnTm pin output
Fixed to low-level output
TOBnBm pin output
Fixed to high-level output
TABnOEm bit = 1, TABnOLm bit = 0 (positive-phase output)
TABnOEBm bit = 1, TABnOLBm bit = 1 (negative-phase output)
TOBnTm pin output
TOBnBm pin output
TABnOEm bit = 1, TABnOLm bit = 0 (positive-phase output)
TABnOEBm bit = 1, TABnOLBm bit = 0 (positive-phase output)
TOBnTm pin output
TOBnBm pin output
TABnOEm bit = 1, TABnOLm bit = 1 (negative-phase output)
TABnOEBm bit = 1, TABnOLBm bit = 1 (negative-phase output)
TOBnTm pin output
TOBnBm pin output

Remark V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

Table 10-1. TOBnTm Pin Output

TABnOLm Bit	TABnOEm Bit	TABnCE Bit	TOBnTm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOBnTm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOBnTm negative-phase output

Remark V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

Table 10-2. TOBnBm Pin Output

TABnOLBm Bit	TABnOEBm Bit	TABnCE Bit	TOBnBm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOBnBm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOBnBm negative-phase output

Remark V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

(6) High-impedance output control registers 00, 01, 10, 11, 20, 21, 30, 31, 40, 41, 50, 51, 60, 61, 70, 71, 80, 81, 90, 91, 100, 101, 110, 111, 120, 121 (HZAyCTL0, HZAyCTL1)

The HZAyCTL0 and HZAyCTL1 registers are 8-bit registers that control the high-impedance state of the output buffer.

These registers can be read or written in 8-bit or 1-bit units. However, the HZAyDCFn bit is a read-only bit and cannot be written.

16-bit access is not possible.

Reset sets these registers to 00H.

The same value can be always rewritten to the HZAyCTLn register by software.

(a) V850E/IG4-H

The relationship between detection factor and the control registers is shown below.

Pins Subject to High-Impedance Control	High-Impedance Control Factor		Control Register
	External Pin	A/D Unit (Comparator)	
TOB0T1 to TOB0T3 outputs TOB0B1 to TOB0B3 outputs	TOB0OFF	–	HZA0CTL0
TOB0T1 to TOB0T3 outputs			HZA5CTL0
TOB0B1 to TOB0B3 outputs			HZA9CTL0
TOB0T1 to TOB0T3 outputs TOB0B1 to TOB0B3 outputs	TOB01OFF	–	HZA4CTL0
TOB0T1 to TOB0T3 outputs			HZA8CTL0
TOB0B1 to TOB0B3 outputs			HZA12CTL0
TOB0T1 to TOB0T3 outputs TOB0B1 to TOB0B3 outputs	–	When the low range reference voltage of ANI00/ANI05 to ANI02/ANI07 input is exceeded (rising edge) or not reached (falling edge)	HZA2CTL0
TOB0T1 to TOB0T3 outputs			HZA6CTL0
TOB0B1 to TOB0B3 outputs			HZA10CTL0
TOB0T1 to TOB0T3 outputs TOB0B1 to TOB0B3 outputs	–	When the full range reference voltage of ANI00/ANI05 to ANI02/ANI07 input is exceeded (rising edge) or not reached (falling edge)	HZA2CTL0
TOB0T1 to TOB0T3 outputs			HZA6CTL1
TOB0B1 to TOB0B3 outputs			HZA10CTL1
TOT21 output	TOT2OFF	–	HZA0CTL1

Caution High-impedance control is performed only when a port pin is set to function as indicated in the above table.

(1/3)

After reset: 00H R/W Address: HZA0CTL0 FFFFF610H, HZA0CTL1 FFFFF611H,
HZA2CTL0 FFFFF650H, HZA2CTL1 FFFFF651H,
HZA4CTL0 FFFFFE00H, HZA5CTL0 FFFFFE08H,
HZA6CTL0 FFFFFE10H, HZA6CTL1 FFFFFE11H,
HZA8CTL0 FFFFFE20H, HZA9CTL0 FFFFFE28H,
HZA10CTL0 FFFFFE30H, HZA10CTL1 FFFFFE31H,
HZA12CTL0 FFFFFE40H

	<7>	<6>	5	4	<3>	<2>	1	<0>
HZAyCTLn	HZAyDCEn	HZAyDCMn	HZAyDCNn	HZAyDCPn	HZAyDCTn	HZAyDCCn	0	HZAyDCFn

(n = 0, 1
y = 0, 2, 4 to 6,
8 to 10, 12)

HZAyDCEn	High-impedance output control
0	Disable high-impedance output control operation. Pins can function as output pins.
1	Enable high-impedance output control operation.

HZAyDCMn	Condition of clearing high-impedance state by HZAyDCCn bit
0	Setting of the HZAyDCCn bit is valid regardless of the external pin[Note] input.
1	Setting of the HZAyDCCn bit is invalid while the external pin[Note] input holds a level detected as abnormal (active level).
Rewrite the HZAyDCMn bit when the HZAyDCEn bit = 0.	

Note HZA0CTL0, HZA5CTL0, HZA9CTL0: TOB0OFF pin
HZA4CTL0, HZA8CTL0,
HZA12CTL0: TOB01OFF pin
HZA0CTL1: TOT2OFF pin
HZA2CTL0, HZA6CTL0, HZA10CTL0: ANI00/ANI05 to ANI02/ANI07 pins
HZA2CTL1, HZA6CTL1, HZA10CTL1: ANI00/ANI05 to ANI02/ANI07 pins

(2/3)

HZAyDCNn	HZAyDCPn	External pin[Note 1] input edge specification
0	0	No valid edge (setting the HZAyDCFn bit by external pin[Note 1] input is prohibited).
0	1	Rising edge of the external pin[Note 1] input is valid (abnormality is detected by rising edge input)[Note 2].
1	0	Falling edge of the external pin[Note 1] input is valid (abnormality is detected by falling edge input)[Note 2].
1	1	Setting prohibited

- Rewrite the HZAyDCNn and HZAyDCPn bits when the HZAyDCEn bit is 0.
- For the edge specification of the INTP03, INTP07, and INTP08 pins, see **21.4.2 (1) External interrupt rising edge specification register 0 (INTR0, INTF0)**.
- The edge of the external pins must be specified starting from the TOB0OFF, TOB01OFF, and TOT2OFF pins. Then the edge of the external pins other than the TOB0OFF, TOB01OFF, and TOT2OFF pins must be specified.
  Otherwise, the undefined edge may be detected when edges of the TOB0OFF, TOB01OFF, and TOT2OFF pins are specified.
- High-impedance output control is performed when the valid edge is input after the operation is enabled (by setting HZAyDCEn bit to 1). If the external pin[Note 1] is at the active level when the operation is enabled, therefore, high-impedance output control is not performed.

Notes 1. HZA0CTL0, HZA5CTL0, HZA9CTL0: TOB0OFF pin
HZA4CTL0, HZA8CTL0,
HZA12CTL0: TOB01OFF pin
HZA0CTL1: TOT2OFF pin
HZA2CTL0, HZA6CTL0, HZA10CTL0: ANI00/ANI05 to ANI02/ANI07 pins
HZA2CTL1, HZA6CTL1, HZA10CTL1: ANI00/ANI05 to ANI02/ANI07 pins

2. To detect the voltage of a comparator exceeding the reference voltage, set the rising edge input. To detect the voltage of a comparator that has not reached the reference voltage, set the falling edge input.

(3/3)

HZAyDCTn	High-impedance output trigger bit
0	No operation
1	Pins are made to go into a high-impedance state by software and the HZAyDCFn bit is set to 1.
	• If an edge indicating abnormality is input to the external pin[Note 2] (which is detected according to the setting of the HZAyDCNn and HZAyDCPn bits), the HZAyDCTn bit is invalid even if it is set to 1. • The HZAyDCTn bit is always 0 when it is read because it is a software-triggered bit. • The HZAyDCTn bit is invalid even if it is set to 1 when the HZAyDCEn bit = 0. • Simultaneously setting the HZAyDCTn and HZAyDCCn bits to 1 is prohibited.

HZAyDCCn	High-impedance output control clear bit
0	No operation
1	Pins that have gone into a high-impedance state are output-enabled by software and the HZAyDCFn bit is cleared to 0.
	• Pins can function as output pins when the HZAyDCM bit = 0, regardless of the status of the external pin[Note]. • If an edge indicating abnormality is input to the external pin[Note] (which is set by the HZAyDCNn and HZAyDCPn bits) when the HZAyDCM bit = 1, the HZAyDCCn bit is invalid even if it is set to 1. • The HZAyDCCn bit is always 0 when it is read. • The HZAyDCCn bit is invalid even if it is set to 1 when the HZAyDCEn bit = 0. • Simultaneously setting the HZAyDCTn and HZAyDCCn bits to 1 is prohibited.

HZAyDCFn	High-impedance output status flag
0	Indicates that output of the pin is enabled. • This bit is cleared to 0 when the HZAyDCEn bit = 0. • This bit is cleared to 0 when the HZAyDCCn bit = 1.
1	Indicates that the pin goes into a high-impedance state. • This bit is set to 1 when the HZAyDCTn bit = 1. • This bit is set to 1 when an edge indicating abnormality is input to the external pin[Note] (which is detected according to the setting of the HZAyDCNn and HZAyDCPn bits).

Note HZA0CTL0, HZA5CTL0, HZA9CTL0: TOB0OFF pin
HZA4CTL0, HZA8CTL0,
HZA12CTL0: TOB01OFF pin
HZA0CTL1: TOT2OFF pin
HZA2CTL0, HZA6CTL0, HZA10CTL0: ANI00/ANI05 to ANI02/ANI07 pins
HZA2CTL1, HZA6CTL1, HZA10CTL1: ANI00/ANI05 to ANI02/ANI07 pins

(b) V850E/IH4-H

The relationship between detection factor and the control registers is shown below.

Pins Subject to High-Impedance Control	High-Impedance Control Factor		Control Register
	External Pin	A/D Unit (Comparator)	
TOB0T1 to TOB0T3 outputs TOB0B1 to TOB0B3 outputs	TOB0OFF	–	HZA0CTL0
TOB0T1 to TOB0T3 outputs			HZA5CTL0
TOB0B1 to TOB0B3 outputs			HZA9CTL0
TOB0T1 to TOB0T3 outputs TOB0B1 to TOB0B3 outputs	TOB01OFF	–	HZA4CTL0
TOB0T1 to TOB0T3 outputs			HZA8CTL0
TOB0B1 to TOB0B3 outputs			HZA12CTL0
TOB0T1 to TOB0T3 outputs TOB0B1 to TOB0B3 outputs	–	When the low range reference voltage of ANI00/ANI05 to ANI02/ANI07 input is exceeded (rising edge) or not reached (falling edge)	HZA2CTL0
TOB0T1 to TOB0T3 outputs			HZA6CTL0
TOB0B1 to TOB0B3 outputs			HZA10CTL0
TOB0T1 to TOB0T3 outputs TOB0B1 to TOB0B3 outputs	–	When the full range reference voltage of ANI00/ANI05 to ANI02/ANI07 input is exceeded (rising edge) or not reached (falling edge)	HZA2CTL0
TOB0T1 to TOB0T3 outputs			HZA6CTL1
TOB0B1 to TOB0B3 outputs			HZA10CTL1
TOT21 output	TOT2OFF	–	HZA0CTL1
TOB1T1 to TOB1T3 outputs TOB1B1 to TOB1B3 outputs	TOB1OFF	–	HZA1CTL0
TOB1B1 to TOB1T3 outputs			HZA5CTL1
TOB1B1 to TOB1B3 outputs			HZA9CTL1
TOB1T1 to TOB1T3 outputs TOB1B1 to TOB1B3 outputs	TOB01OFF	–	HZA4CTL1
TOB1T1 to TOB1T3 outputs			HZA8CTL1
TOB1B1 to TOB1B3 outputs			HZA12CTL1
TOB1T1 to TOB1T3 outputs TOB1B1 to TOB1B3 outputs	–	When the low range reference voltage of ANI10/ANI15 to ANI12/ANI17 input is exceeded (rising edge) or not reached (falling edge)	HZA3CTL0
TOB1B1 to TOB1B3 outputs			HZA7CTL0
TOB1B1 to TOB1B3 outputs			HZA11CTL0
TOB1T1 to TOB1T3 outputs TOB1B1 to TOB1B3 outputs	–	When the full range reference voltage of ANI10/ANI15 to ANI12/ANI17 input is exceeded (rising edge) or not reached (falling edge)	HZA3CTL1
TOB1T1 to TOB1T3 outputs			HZA7CTL1
TOB1B1 to TOB1B3 outputs			HZA11CTL1
TOT31 output	TOT3OFF	–	HZA1CTL1

Caution High-impedance control is performed only when a port pin is set to function as indicated in the above table.

(1/3)

After reset: 00H R/W Address:

HZA0CTL0	FFFFF610H,	HZA0CTL1	FFFFF611H,
HZA1CTL0	FFFFF618H,	HZA1CTL1	FFFFF619H,
HZA2CTL0	FFFFF650H,	HZA2CTL1	FFFFF651H,
HZA3CTL0	FFFFF658H,	HZA3CTL1	FFFFF659H
HZA4CTL0	FFFFFE00H,	HZA4CTL1	FFFFFE01H,
HZA5CTL0	FFFFFE08H,	HZA5CTL1	FFFFFE09H,
HZA6CTL0	FFFFFE10H,	HZA6CTL1	FFFFFE11H,
HZA7CTL0	FFFFFE18H,	HZA7CTL1	FFFFFE19H,
HZA8CTL0	FFFFFE20H,	HZA8CTL1	FFFFFE21H,
HZA9CTL0	FFFFFE28H,	HZA9CTL1	FFFFFE29H,
HZA10CTL0	FFFFFE30H,	HZA10CTL1	FFFFFE31H,
HZA11CTL0	FFFFFE38H,	HZA11CTL1	FFFFFE39H,
HZA12CTL0	FFFFFE40H,	HZA12CTL1	FFFFFE41H

	<7>	<6>	5	4	<3>	<2>	1	<0>
HZAyCTLn	HZAyDCEn	HZAyDCMn	HZAyDCNn	HZAyDCPn	HZAyDCTn	HZAyDCCn	0	HZAyDCFn

(n = 0, 1
y = 0 to 12)

HZAyDCEn	High-impedance output control
0	Disable high-impedance output control operation. Pins can function as output pins.
1	Enable high-impedance output control operation.

HZAyDCMn	Condition of clearing high-impedance state by HZAyDCCn bit
0	Setting of the HZAyDCCn bit is valid regardless of the external pin[Note] input.
1	Setting of the HZAyDCCn bit is invalid while the external pin[Note] input holds a level detected as abnormal (active level).
Rewrite the HZAyDCMn bit when the HZAyDCEn bit = 0.	

Note HZA0CTL0, HZA5CTL0, HZA9CTL0: TOB0OFF pin
HZA1CTL0, HZA5CTL1, HZA9CTL1: TOB1OFF pin
HZA4CTL0, HZA4CTL1, HZA8CTL0,
HZA8CTL1, HZA12CTL0, HZA12CTL1: TOB01OFF pin
HZA0CTL1: TOT2OFF pin
HZA1CTL1: TOT3OFF pin
HZA2CTL0, HZA6CTL0, HZA10CTL0: ANI00/ANI05 to ANI02/ANI07 pins
HZA2CTL1, HZA6CTL1, HZA10CTL1: ANI00/ANI05 to ANI02/ANI07 pins
HZA3CTL0, HZA7CTL0, HZA11CTL0: ANI10/ANI15 to ANI12/ANI17 pins
HZA3CTL1, HZA7CTL1, HZA11CTL1: ANI10/ANI15 to ANI12/ANI17 pins

(2/3)

HZAyDCNn	HZAyDCPn	External pin[Note 1] input edge specification
0	0	No valid edge (setting the HZAyDCFn bit by external pin[Note 1] input is prohibited).
0	1	Rising edge of the external pin[Note 1] input is valid (abnormality is detected by rising edge input)[Note 2].
1	0	Falling edge of the external pin[Note 1] input is valid (abnormality is detected by falling edge input)[Note 2].
1	1	Setting prohibited

- Rewrite the HZAyDCNn and HZAyDCPn bits when the HZAyDCEn bit is 0.
- For the edge specification of the INTP03, INTP05, INTP07, INTP08, and INTP10 pins, see **19.4.2 (1) External interrupt rising edge specification register 0 (INTR0, INTF0)**.
- The edge of the external pins must be specified starting from the TOB0OFF, TOB1OFF, TOB01OFF, TOT2OFF, and TOT3OFF pins. Then the edge of the external pins other than the TOB0OFF, TOB1OFF, TOB01OFF, TOT2OFF, and TOT3OFF pins must be specified.
  Otherwise, the undefined edge may be detected when edges of the TOB0OFF, TOB1OFF, TOB01OFF, TOT2OFF, and TOT3OFF pins are specified.
- High-impedance output control is performed when the valid edge is input after the operation is enabled (by setting HZAyDCEn bit to 1). If the external pin[Note 1] is at the active level when the operation is enabled, therefore, high-impedance output control is not performed.

Notes 1. HZA0CTL0, HZA5CTL0, HZA9CTL0: TOB0OFF pin
HZA1CTL0, HZA5CTL1, HZA9CTL1: TOB1OFF pin
HZA4CTL0, HZA4CTL1, HZA8CTL0,
HZA8CTL1, HZA12CTL0, HZA12CTL1: TOB01OFF pin
HZA0CTL1: TOT2OFF pin
HZA1CTL1: TOT3OFF pin
HZA2CTL0, HZA6CTL0, HZA10CTL0: ANI00/ANI05 to ANI02/ANI07 pins
HZA2CTL1, HZA6CTL1, HZA10CTL1: ANI00/ANI05 to ANI02/ANI07 pins
HZA3CTL0, HZA7CTL0, HZA11CTL0: ANI10/ANI15 to ANI12/ANI17 pins
HZA3CTL1, HZA7CTL1, HZA11CTL1: ANI10/ANI15 to ANI12/ANI17 pins

2. To detect the voltage of a comparator exceeding the reference voltage, set the rising edge input. To detect the voltage of a comparator that has not reached the reference voltage, set the falling edge input.

(3/3)

HZAyDCTn	High-impedance output trigger bit
0	No operation
1	Pins are made to go into a high-impedance state by software and the HZAyDCFn bit is set to 1.

- If an edge indicating abnormality is input to the external pin[Note 2] (which is detected according to the setting of the HZAyDCNn and HZAyDCPn bits), the HZAyDCTn bit is invalid even if it is set to 1.
- The HZAyDCTn bit is always 0 when it is read because it is a software-triggered bit.
- The HZAyDCTn bit is invalid even if it is set to 1 when the HZAyDCEn bit = 0.
- Simultaneously setting the HZAyDCTn and HZAyDCCn bits to 1 is prohibited.

HZAyDCCn	High-impedance output control clear bit
0	No operation
1	Pins that have gone into a high-impedance state are output-enabled by software and the HZAyDCFn bit is cleared to 0.

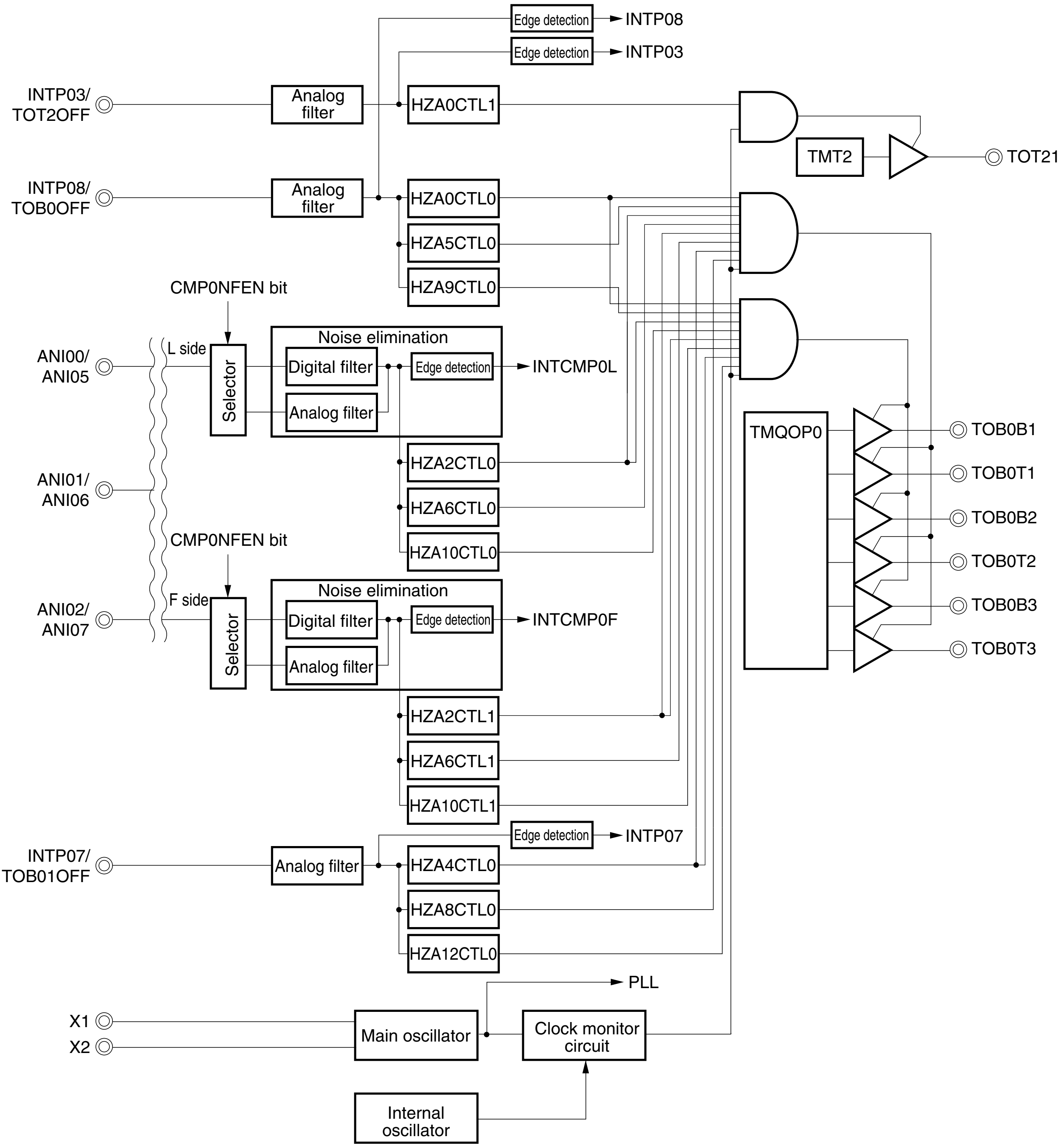
- Pins can function as output pins when the HZAyDCM bit = 0, regardless of the status of the external pin[Note].
- If an edge indicating abnormality is input to the external pin[Note] (which is set by the HZAyDCNn and HZAyDCPn bits) when the HZAyDCM bit = 1, the HZAyDCCn bit is invalid even if it is set to 1.
- The HZAyDCCn bit is always 0 when it is read.
- The HZAyDCCn bit is invalid even if it is set to 1 when the HZAyDCEn bit = 0.
- Simultaneously setting the HZAyDCTn and HZAyDCCn bits to 1 is prohibited.

HZAyDCFn	High-impedance output status flag
0	Indicates that output of the pin is enabled. • This bit is cleared to 0 when the HZAyDCEn bit = 0. • This bit is cleared to 0 when the HZAyDCCn bit = 1.
1	Indicates that the pin goes into a high-impedance state. • This bit is set to 1 when the HZAyDCTn bit = 1. • This bit is set to 1 when an edge indicating abnormality is input to the external pin[Note] (which is detected according to the setting of the HZAyDCNn and HZAyDCPn bits).

Note HZA0CTL0, HZA5CTL0, HZA9CTL0: TOB0OFF pin
HZA1CTL0, HZA5CTL1, HZA9CTL1: TOB1OFF pin
HZA4CTL0, HZA4CTL1, HZA8CTL0,
HZA8CTL1, HZA12CTL0, HZA12CTL1: TOB01OFF pin
HZA0CTL1: TOT2OFF pin
HZA1CTL1: TOT3OFF pin
HZA2CTL0, HZA6CTL0, HZA10CTL0: ANI00/ANI05 to ANI02/ANI07 pins
HZA2CTL1, HZA6CTL1, HZA10CTL1: ANI00/ANI05 to ANI02/ANI07 pins
HZA3CTL0, HZA7CTL0, HZA11CTL0: ANI10/ANI15 to ANI12/ANI17 pins
HZA3CTL1, HZA7CTL1, HZA11CTL1: ANI10/ANI15 to ANI12/ANI17 pins

Figure 10-4. High-Impedance Output Controller Configuration (1/2)

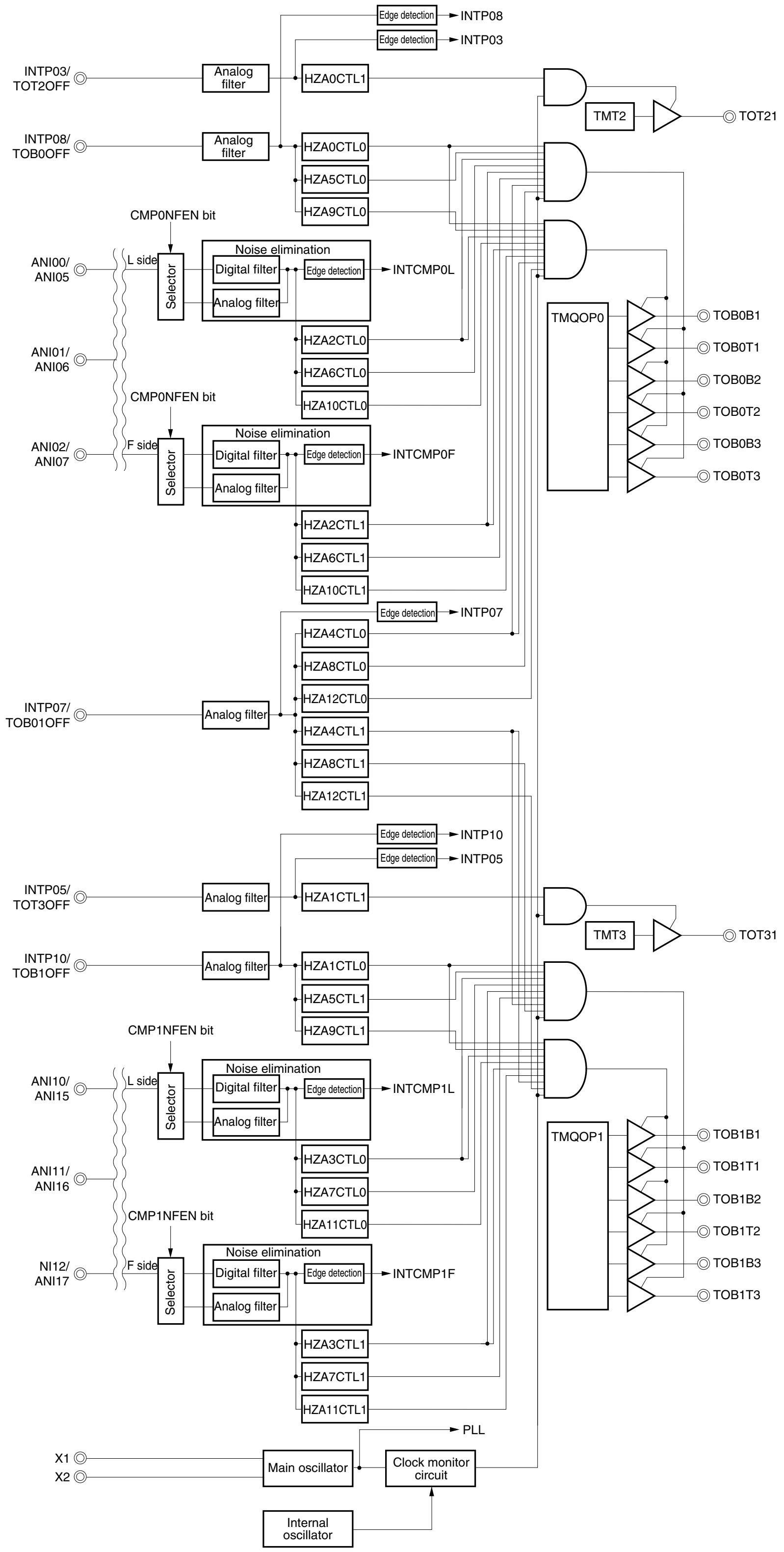
(a) V850E/IG4-H



Remark When referring to **Figure 10-4**, also refer to **Figures 12-3** and **12-4**.

Figure 10-4. High-Impedance Output Controller Configuration (2/2)

(b) V850E/IH4-H



Remark When referring to **Figure 10-4**, also refer to **Figures 12-3** and **12-4**.

(c) Setting procedure

(i) Setting of high-impedance control operation

<1> Set the HZAyDCMn, HZAyDCNn, and HZAyDCPn bits.
<2> Set the HZAyDCEn bit to 1 (enable high-impedance control).

(ii) Changing setting after enabling high-impedance control operation

<1> Clear the HZAyDCEn bit to 0 (to stop the high-impedance control operation).
<2> Change the setting of the HZAyDCMn, HZAyDCNn, and HZAyDCPn bits.
<3> Set the HZAyDCEn bit to 1 (to enable the high-impedance control operation again).

(iii) Resuming output when pins are in high-impedance state

If the HZAyDCMn bit is 1, set the HZAyDCCn bit to 1 to clear the high-impedance state after the valid edge of the external pin[Note] is detected. However, the high-impedance state cannot be cleared unless this bit is set while the input level of the external pin[Note] is inactive.

<1> Set the HZAyDCCn bit to 1 (command signal to clear the high-impedance state).
<2> Read the HZAyDCFn bit and check the flag status.
<3> Return to <1> if the HZAyDCFn bit is 1. The input level of the external pin[Note] must be checked. The pin can function as an output pin if the HZAyDCFn bit is 0.

(iv) To make the pin to go into a high-impedance state by software

The HZAyDCTn bit must be set to 1 by software to make the pin to go into a high-impedance state while the input level of the external pin[Note] is inactive. The following procedure is an example in which the setting is not dependent upon the setting of the HZAyDCMn bit.

<1> Set the HZAyDCTn bit to 1 (high-impedance output command).
<2> Read the HZAyDCFn bit to check the flag status.
<3> Return to <1> if the HZAyDCFn bit is 0. The input level of the external pin[Note] must be checked. The pin is in a high-impedance state if the HZAyDCFn bit is 1.

However, if the external pin[Note] is not used with the HZAyDCPn bit and HZAyDCNn bit cleared to 0, the pin goes into a high-impedance state when the HZAyDCTn bit is set to 1.

Note
- V850E/IG4-H
  HZA0CTL0, HZA5CTL0, HZA9CTL0: TOB0OFF pin
  HZA4CTL0, HZA8CTL0, HZA12CTL0: TOB01OFF pin
  HZA0CTL1: TOT2OFF pin
  HZA2CTL0, HZA6CTL0, HZA10CTL0: ANI00/ANI05 to ANI02/ANI07 pins
  HZA2CTL1, HZA6CTL1, HZA10CTL1: ANI00/ANI05 to ANI02/ANI07 pins
- V850E/IH4-H
  HZA0CTL0, HZA5CTL0, HZA9CTL0: TOB0OFF pin
  HZA1CTL0, HZA5CTL1, HZA9CTL1: TOB1OFF pin
  HZA4CTL0, HZA4CTL1, HZA8CTL0,
  HZA8CTL1, HZA12CTL0, HZA12CTL1: TOB01OFF pin
  HZA0CTL1: TOT2OFF pin
  HZA1CTL1: TOT3OFF pin
  HZA2CTL0, HZA6CTL0, HZA10CTL0: ANI00/ANI05 to ANI02/ANI07 pins
  HZA2CTL1, HZA6CTL1, HZA10CTL1: ANI00/ANI05 to ANI02/ANI07 pins
  HZA3CTL0, HZA7CTL0, HZA11CTL0: ANI10/ANI15 to ANI12/ANI17 pins
  HZA3CTL1, HZA7CTL1, HZA11CTL1: ANI10/ANI15 to ANI12/ANI17 pins

10.4 Operation

10.4.1 System outline

(1) Outline of 6-phase PWM output

The 6-phase PWM output mode is used to generate a 6-phase PWM output waveform, by using TABn and the TMQn option in combination.

The 6-phase PWM output mode is enabled by setting the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits of TABn to “111”.

One 16-bit counter and four 16-bit compare registers of TABn are used to generate a basic 3-phase wave.

The functions of the compare registers are as follows.

TAAn can perform a tuning operation with TABn to start a conversion trigger source for A/D converters 0 and 1.

Remark V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

Compare Register	Function	Settable Range
TABnCCR0 register	Setting of cycle	0002H ≤ m ≤ FFFEH
TABnCCR1 register	Specifying output width of phase U	0000H ≤ i ≤ m + 1
TABnCCR2 register	Specifying output width of phase V	0000H ≤ j ≤ m + 1
TABnCCR3 register	Specifying output width of phase W	0000H ≤ k ≤ m + 1

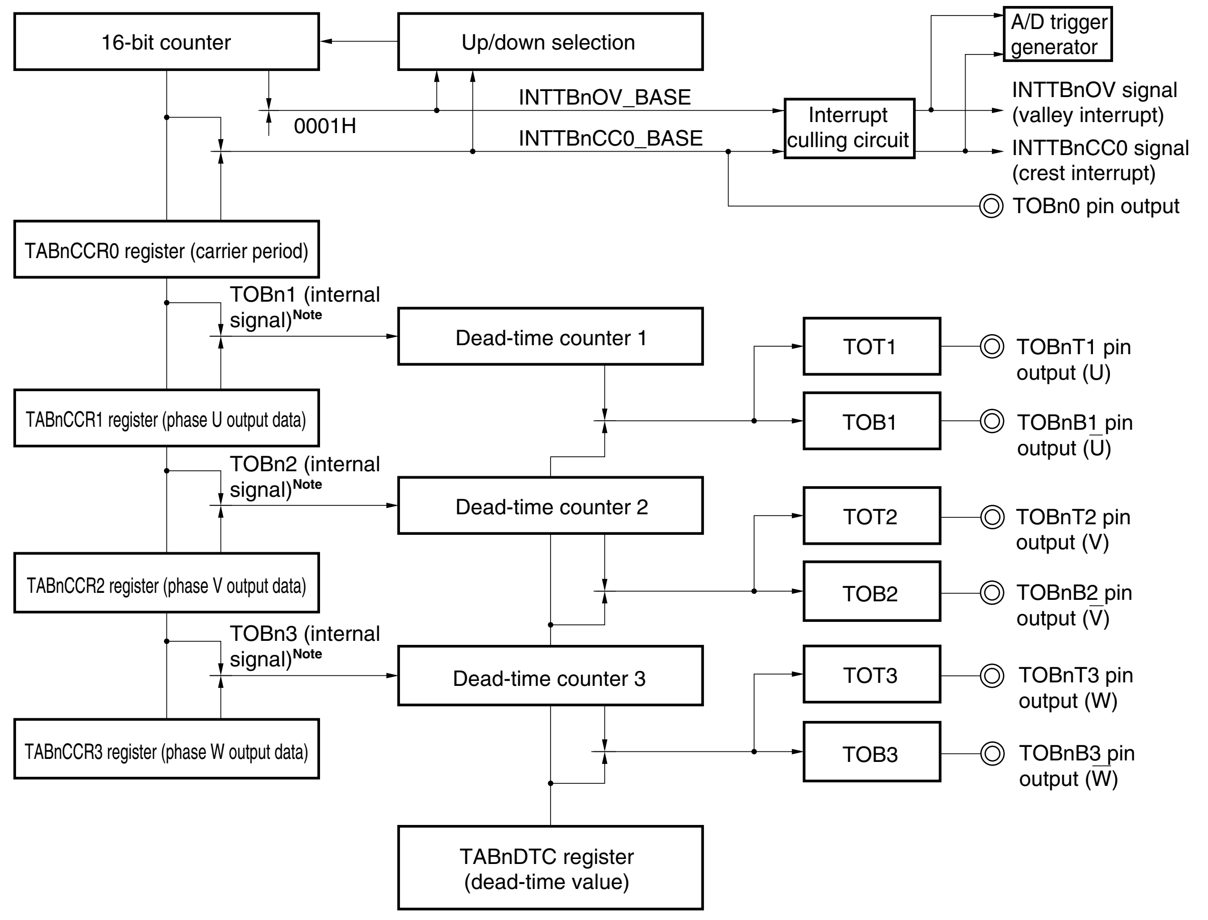
Remark m = Set value of TABnCCR0 register
i = Set value of TABnCCR1 register
j = Set value of TABnCCR2 register
k = Set value of TABnCCR3 register

A dead-time interval is generated from the basic 3-phase wave generated by using three 10-bit dead-time counters and one compare register to create a wave with a reverse phase to that of the basic 3-phase wave. Then a 6-phase PWM output waveform (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$) is generated.

The 16-bit counter for generating the basic 3-phase wave counts up or down. After the operation has been started, this counter counts up. When its count value matches the cycle set to the TABnCCR0 register, the counter starts counting down. When the count value matches 0001H, the counter counts up again. This means that a value two times higher than the value set to the TABnCCR0 register + 1 is the carrier cycle.

10-bit dead-time counters 1 to 3 that generate the dead-time interval count up. Therefore, the value set to the TABn dead-time compare register (TABnDTC) is used as a dead-time value as is. Because three counters are used, dead time can be generated independently in phases U, V, and W. However, because there is only one register that specifies a dead-time value (TABnDTC), the same dead-time value is used in the three phases.

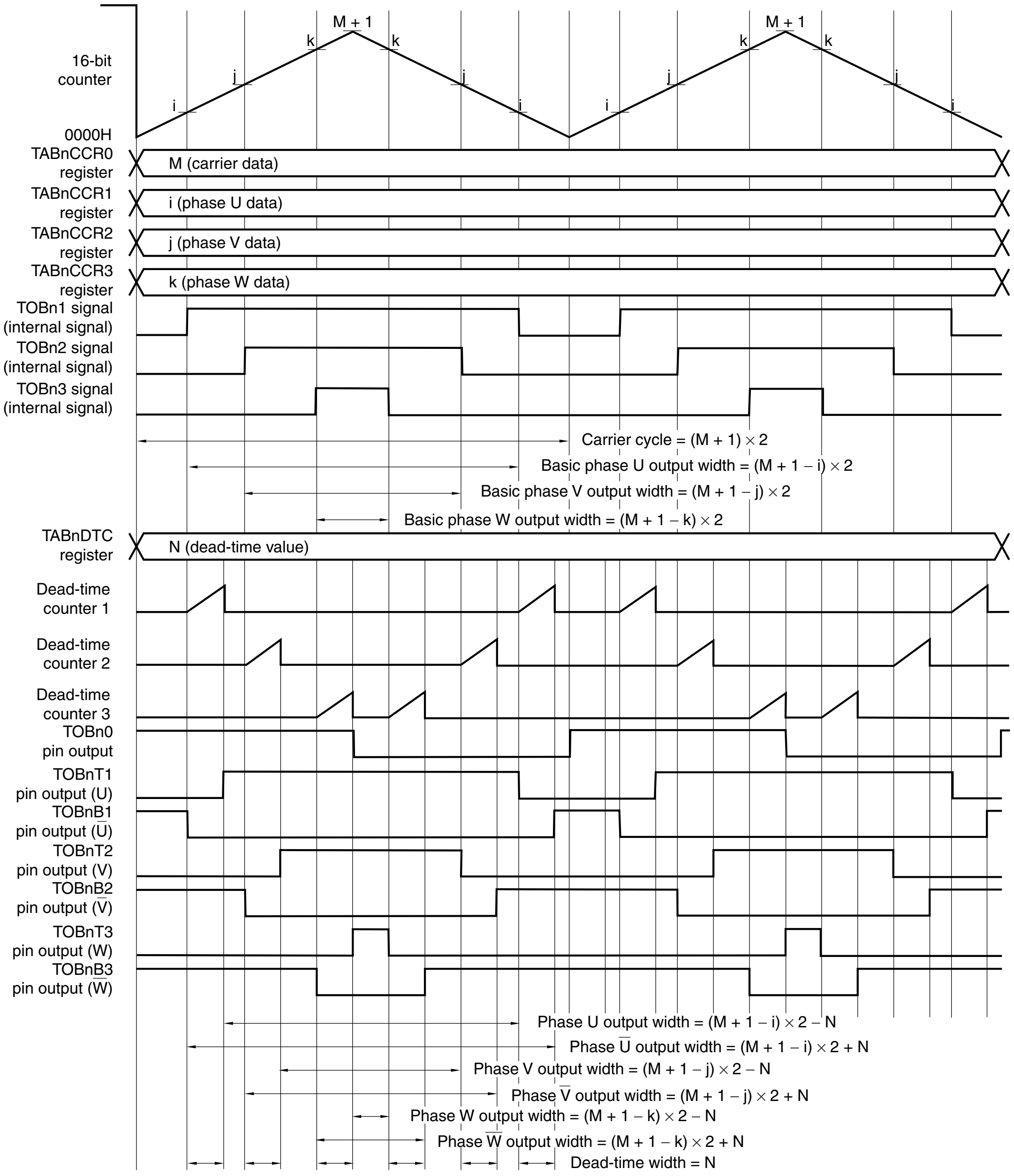
Figure 10-5. Outline of 6-Phase PWM Output Mode


16-bit counter
Up/down selection
A/D trigger generator
INTTBnOV_BASE
0001H
INTTBnCC0_BASE
Interrupt culling circuit
INTTBnOV signal (valley interrupt)
INTTBnCC0 signal (crest interrupt)
TOBn0 pin output
TABnCCR0 register (carrier period)
TOBn1 (internal signal)Note
Dead-time counter 1
TOT1
TOBnT1 pin output (U)
TOB1
TOBnB1 pin output ($\overline{U}$)
TABnCCR1 register (phase U output data)
TOBn2 (internal signal)Note
Dead-time counter 2
TOT2
TOBnT2 pin output (V)
TOB2
TOBnB2 pin output ($\overline{V}$)
TABnCCR2 register (phase V output data)
TOBn3 (internal signal)Note
Dead-time counter 3
TOT3
TOBnT3 pin output (W)
TOB3
TOBnB3 pin output ($\overline{W}$)
TABnCCR3 register (phase W output data)
TABnDTC register (dead-time value)

Note TOBn1, TOBn2, and TOBn3 function alternately as output pins.

Remark V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

Figure 10-6. Timing Chart of 6-Phase PWM Output Mode


M + 1
k
k
j
j
i
i
M + 1
16-bit counter
0000H
TABnCCR0 register
M (carrier data)
TABnCCR1 register
i (phase U data)
TABnCCR2 register
j (phase V data)
TABnCCR3 register
k (phase W data)
TOBn1 signal (internal signal)
TOBn2 signal (internal signal)
TOBn3 signal (internal signal)
Carrier cycle = (M + 1) × 2
Basic phase U output width = (M + 1 − i) × 2
Basic phase V output width = (M + 1 − j) × 2
Basic phase W output width = (M + 1 − k) × 2
TABnDTC register
N (dead-time value)
Dead-time counter 1
Dead-time counter 2
Dead-time counter 3
TOBn0 pin output
TOBnT1 pin output (U)
TOBnB1 pin output ($\overline{U}$)
TOBnT2 pin output (V)
TOBnB2 pin output ($\overline{V}$)
TOBnT3 pin output (W)
TOBnB3 pin output ($\overline{W}$)
Phase U output width = (M + 1 − i) × 2 − N
Phase $\overline{U}$ output width = (M + 1 − i) × 2 + N
Phase V output width = (M + 1 − j) × 2 − N
Phase $\overline{V}$ output width = (M + 1 − j) × 2 + N
Phase W output width = (M + 1 − k) × 2 − N
Phase $\overline{W}$ output width = (M + 1 − k) × 2 + N
Dead-time width = N

Cautions 1. Set the value "M" of the TABnCCR0 register in a range of 0002H ≤ M ≤ FFFEH in the 6-phase PWM output mode.

2. Only a value of up to "M + 1" can be set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers.

3. The output is 100% if "0000H" is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers. The output is 0% if "M + 1" is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers.

The output (duty 50%) rises at the crest (M + 1) of the 16-bit counter and falls at the valley (0000H) if "M + 2" or higher is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers.

4. If the value calculated using an equation for the output width of the positive phase side of the U, V, or W phase (such as ((M + 1 − i) × 2 − N)) is 0 or less, the output converges at 0 (0%). If the range of the calculated value is from ((M + 1) × 2 − N) to ((M + 1) × 2), the output converges at ((M + 1) × 2) (100%).

Remark V850E/IG4-H: n = 0, V850E/IH4-H: n = 0, 1

(2) Interrupt requests

Two types of interrupt requests are available: the INTTBnCC0 (crest interrupt) signal and INTTBnOV (valley interrupt) signal.

The INTTBnCC0 and INTTBnOV signals can be culled by using the TABnOPT1 register.

For details of culling interrupts, see **10.4.3 Interrupt culling function**.

- INTTBnCC0 (crest interrupt) signal: Interrupt signal indicating matching between the value of the 16-bit counter that counts up and the value of the TABnCCR0 register
- INTTBnOV (valley interrupt) signal: Interrupt signal indicating matching between the value of the 16-bit counter that counts down and the value 0001H

(3) Rewriting registers during timer operation

The following registers have a buffer register and can be rewritten in the anytime rewriting mode, batch rewrite mode, or intermittent batch rewrite mode.

Related Unit	Register
Timer AAn	TAAn capture/compare register 0 (TAAnCCR0) TAAn capture/compare register 1 (TAAnCCR1)
Timer ABn	TABn capture/compare register 0 (TABnCCR0) TABn capture/compare register 1 (TABnCCR1) TABn capture/compare register 2 (TABnCCR2) TABn capture/compare register 3 (TABnCCR3)
Timer Qn option	TABn option register 1 (TABnOPT1)

Remark V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

For details of the transfer function of the compare register, see **10.4.4 Operation to rewrite register with transfer function**.

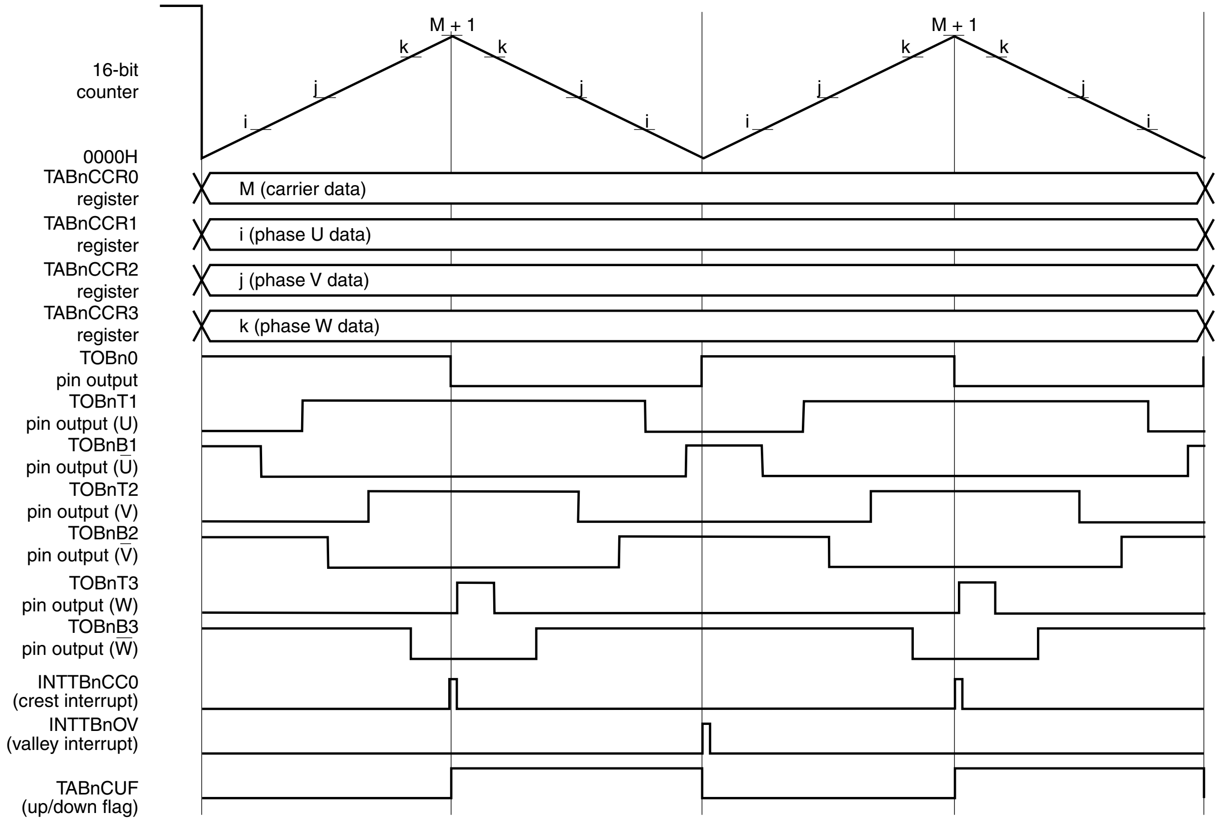
(4) Counting-up/-down operation of 16-bit counter

The operation status of the 16-bit counter can be checked by using the TABnCUF bit of TABn option register 0 (TABnOPT0).

Status of TABnCUF Bit	Status of 16-bit Counter	Range of 16-bit Counter Value
TABnCUF bit = 0	Counting up	0000H – m
TABnCUF bit = 1	Counting down	(m+1) – 0001H

Remarks 1. m = Set value of TABnCCR0 register
2. V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

Figure 10-7. Interrupt and Up/Down Flag



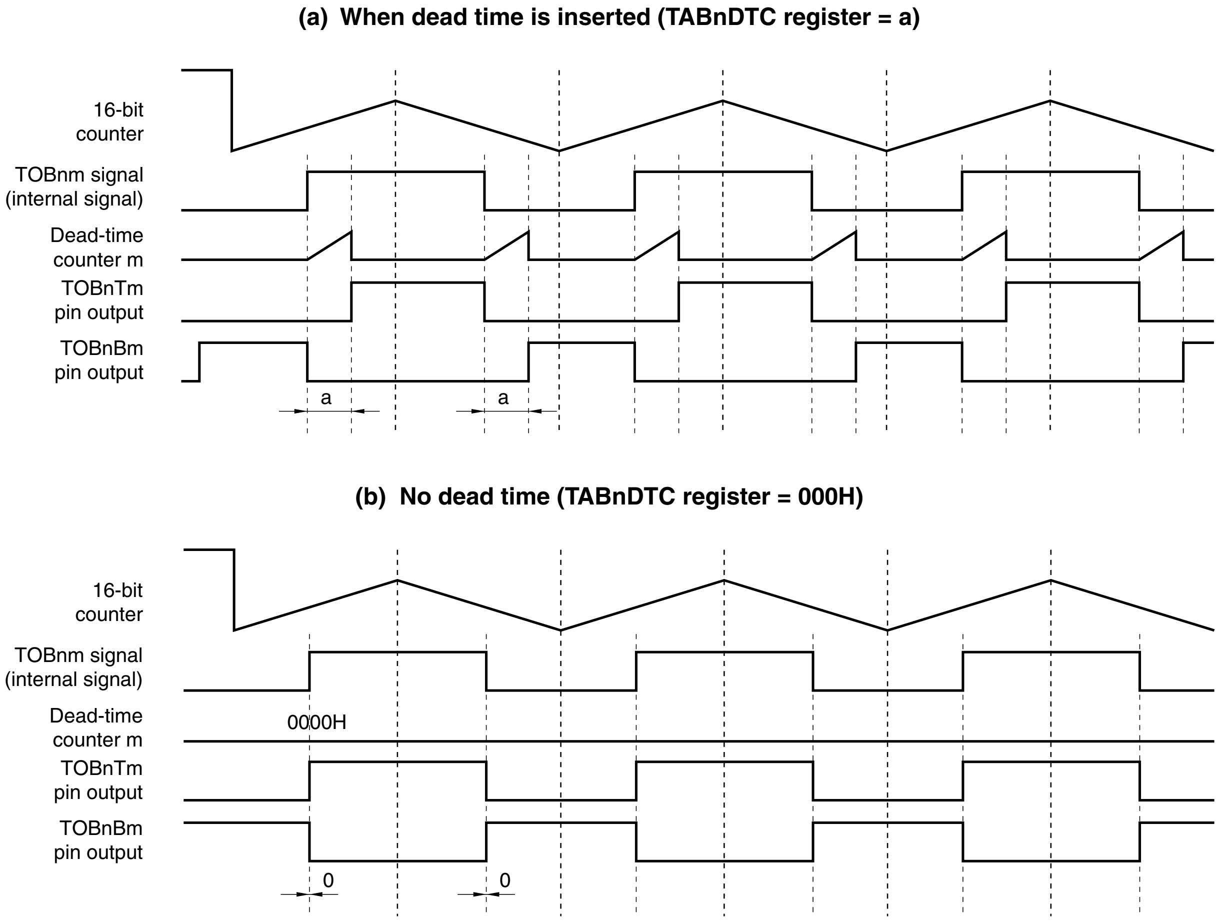
Remark V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

10.4.2 Dead-time control (generation of negative-phase wave signal)

(1) Dead-time control mechanism

In the 6-phase PWM output mode, compare registers 1 to 3 (TABnCCR1, TABnCCR2, and TABnCCR3) are used to set the duty factor, and compare register 0 (TABnCCR0) is used to set the cycle. By setting these four registers and by starting the operation of TAB, three types of PWM output waves (basic 3-phase waves) with a variable duty factor are generated. These three PWM output waves are input to the timer Qn option unit (TMQOPn) and their inverted signal with dead-time is created to generate three sets of (six) PWM waves. The TMQOPn unit consists of three 10-bit counters (dead-time counters 1 to 3) that operate in synchronization with the count clock of TABn, and a TABn dead-time compare register (TABnDTC) that specifies dead time. If “a” is set to the TABnDTC register, the dead-time value is “a”, and interval “a” is created between a positive-phase wave and a negative-phase wave.

Figure 10-8. PWM Output Waveform with Dead Time (1)


(a) When dead time is inserted (TABnDTC register = a)
16-bit counter
TOBnm signal (internal signal)
Dead-time counter m
TOBnTm pin output
TOBnBm pin output
a
a
(b) No dead time (TABnDTC register = 000H)
16-bit counter
TOBnm signal (internal signal)
Dead-time counter m
0000H
TOBnTm pin output
TOBnBm pin output
0
0

Remark V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

(2) PWM output of 0%/100%

The V850E/IG4-H and V850E/IH4-H are capable of 0% waveform output and 100% waveform output for PWM output.

A low level is continuously output from TOBnTm pin as the 0% waveform output. A high level is continuously output from TOBnTm pin as the 100% waveform output.

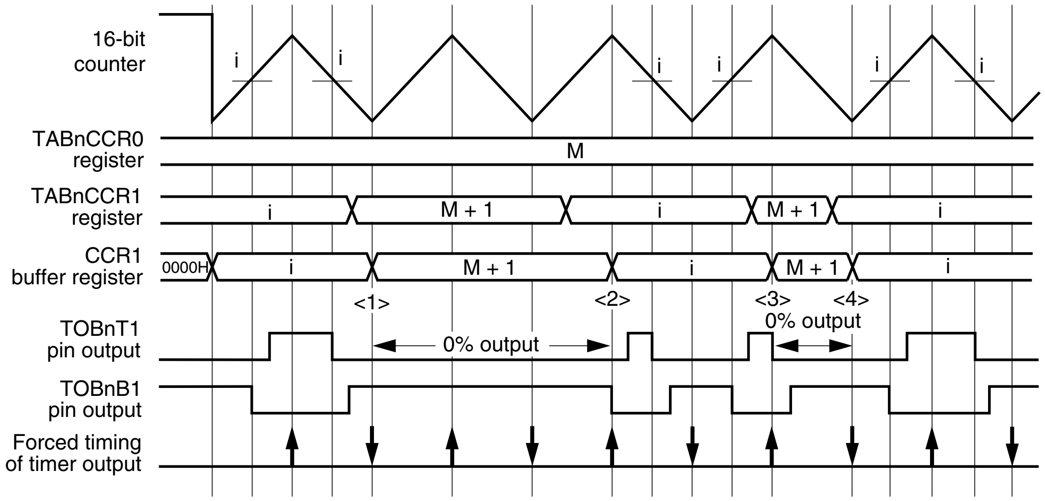
The 0% waveform is output by setting the TABnCCRm register to “M + 1” when the TABnCCR0 register = M.

The 100% waveform is output by setting the TABnCCRm register to “0000H”.

Rewriting the TABnCCRm register is enabled while the timer is operating, and 0% waveform output or 100% waveform output can be selected at the point of the crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV).

Remark V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

Figure 10-9. 0% PWM Output Waveform (With Dead Time)



<1> 0% output is selected by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This produces the 0% output.

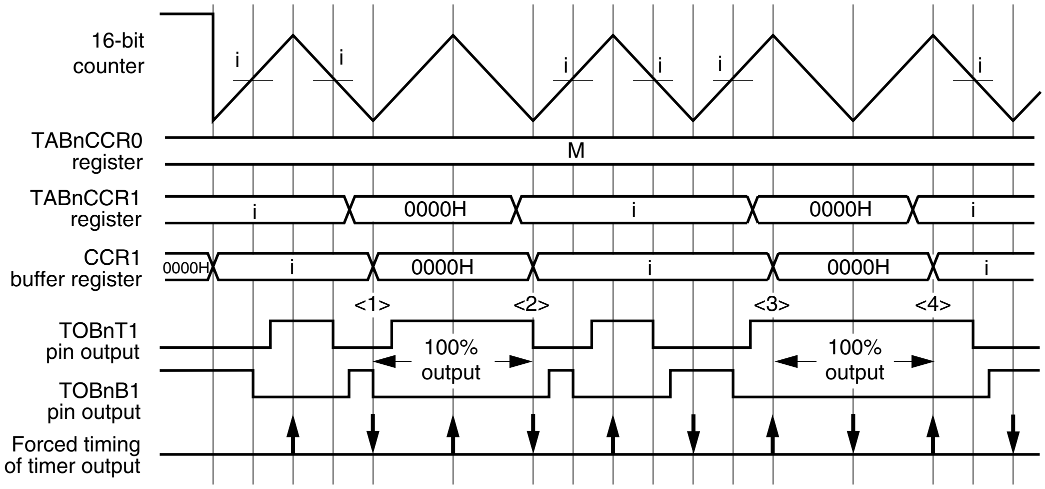
<2> 0% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 0% output.

<3> 0% output is selected by the crest interrupt (with a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output, but lowering the timer output takes precedence when the value of the TABnCCRm register matches the value of the 16-bit counter. As a result, the 0% wave is output.

<4> 0% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 0% output.

Remark ↑ means forced raising and ↓ means forced lowering.

Figure 10-10. 100% PWM Output Waveform (With Dead Time)



<1> 100% output is selected by the valley interrupt (with a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output, but raising the timer output takes precedence when the value of the TABnCCRm register matches the value of the 16-bit counter. As a result, the 100% output is produced.

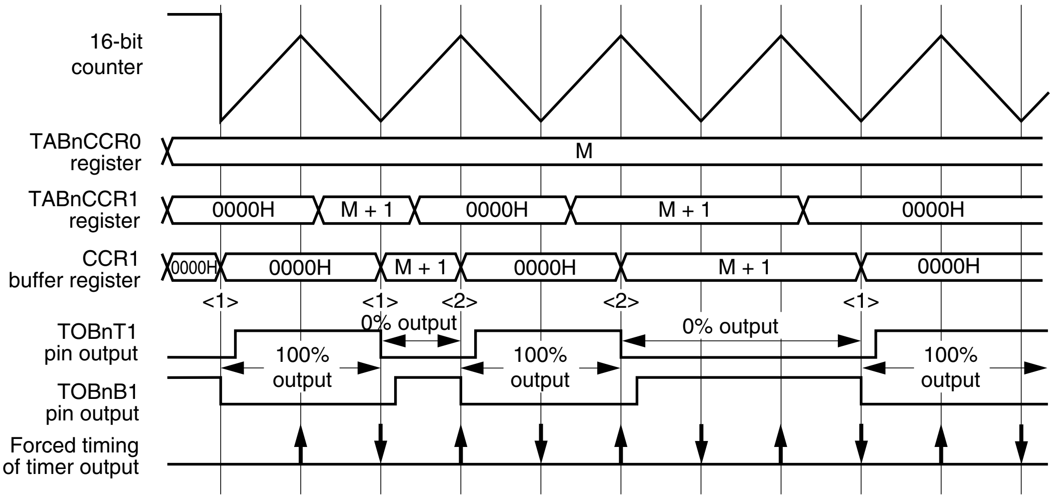
<2> 100% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 100% output.

<3> 100% output is selected by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This produces the 100% output.

<4> 100% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 100% output.

Remark ↑ means forced raising and ↓ means forced lowering.

Figure 10-11. PWM Output Waveform from 0% to 100% and from 100% to 0% (With Dead Time)



<1> The valley interrupt selects 100% ←→ 0% or 0% ←→ 100% output.
Output can be selected from 100% ←→ 0% or 0% ←→ 100% immediately after the timer has been started.

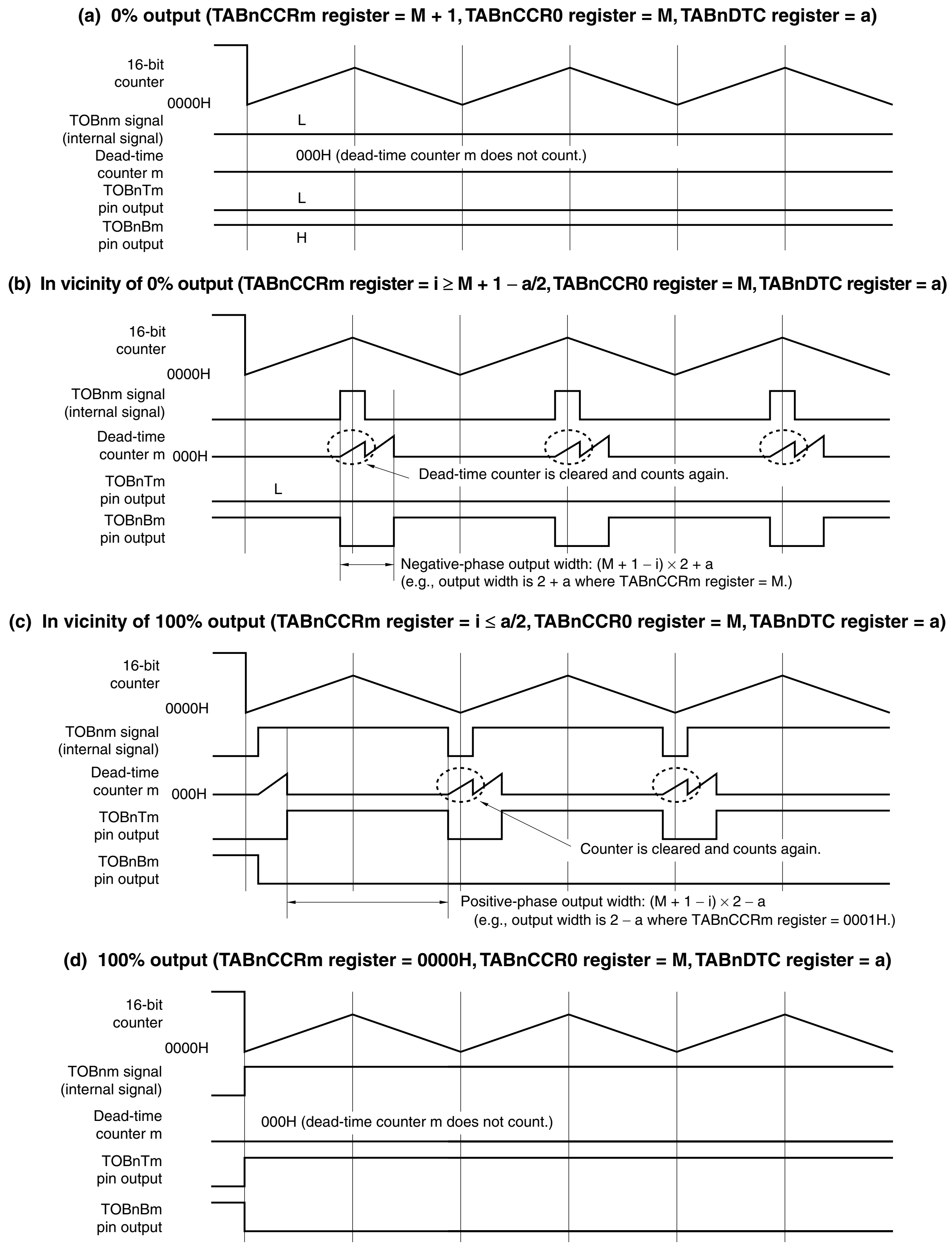
<2> The crest interrupt selects 100% ←→ 0% output.
The crest interrupt selects 100% → 0% output by using the timer output forced raising function and by a match between the 16-bit counter value and the TABnCCR0 register value.

(3) Output wave in vicinity of 0% and 100% output

If an interrupt is generated because the value of the 16-bit counter matches the value of the compare register while dead time is being counted, the dead-time counter is cleared and starts its count operation again.

The output waveform of dead-time control in the vicinity of 0% and 100% output is shown below.

Figure 10-12. PWM Output Waveform with Dead Time (2)


(a) 0% output (TABnCCRm register = M + 1, TABnCCR0 register = M, TABnDTC register = a)
16-bit counter
0000H
TOBnm signal (internal signal)
L
Dead-time counter m
000H (dead-time counter m does not count.)
TOBnTm pin output
L
TOBnBm pin output
H
(b) In vicinity of 0% output (TABnCCRm register = i ≥ M + 1 − a/2, TABnCCR0 register = M, TABnDTC register = a)
16-bit counter
0000H
TOBnm signal (internal signal)
Dead-time counter m
000H
Dead-time counter is cleared and counts again.
TOBnTm pin output
L
TOBnBm pin output
Negative-phase output width: (M + 1 − i) × 2 + a
(e.g., output width is 2 + a where TABnCCRm register = M.)
(c) In vicinity of 100% output (TABnCCRm register = i ≤ a/2, TABnCCR0 register = M, TABnDTC register = a)
16-bit counter
0000H
TOBnm signal (internal signal)
Dead-time counter m
000H
TOBnTm pin output
Counter is cleared and counts again.
TOBnBm pin output
Positive-phase output width: (M + 1 − i) × 2 − a
(e.g., output width is 2 − a where TABnCCRm register = 0001H.)
(d) 100% output (TABnCCRm register = 0000H, TABnCCR0 register = M, TABnDTC register = a)
16-bit counter
0000H
TOBnm signal (internal signal)
Dead-time counter m
000H (dead-time counter m does not count.)
TOBnTm pin output
TOBnBm pin output

Remark V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

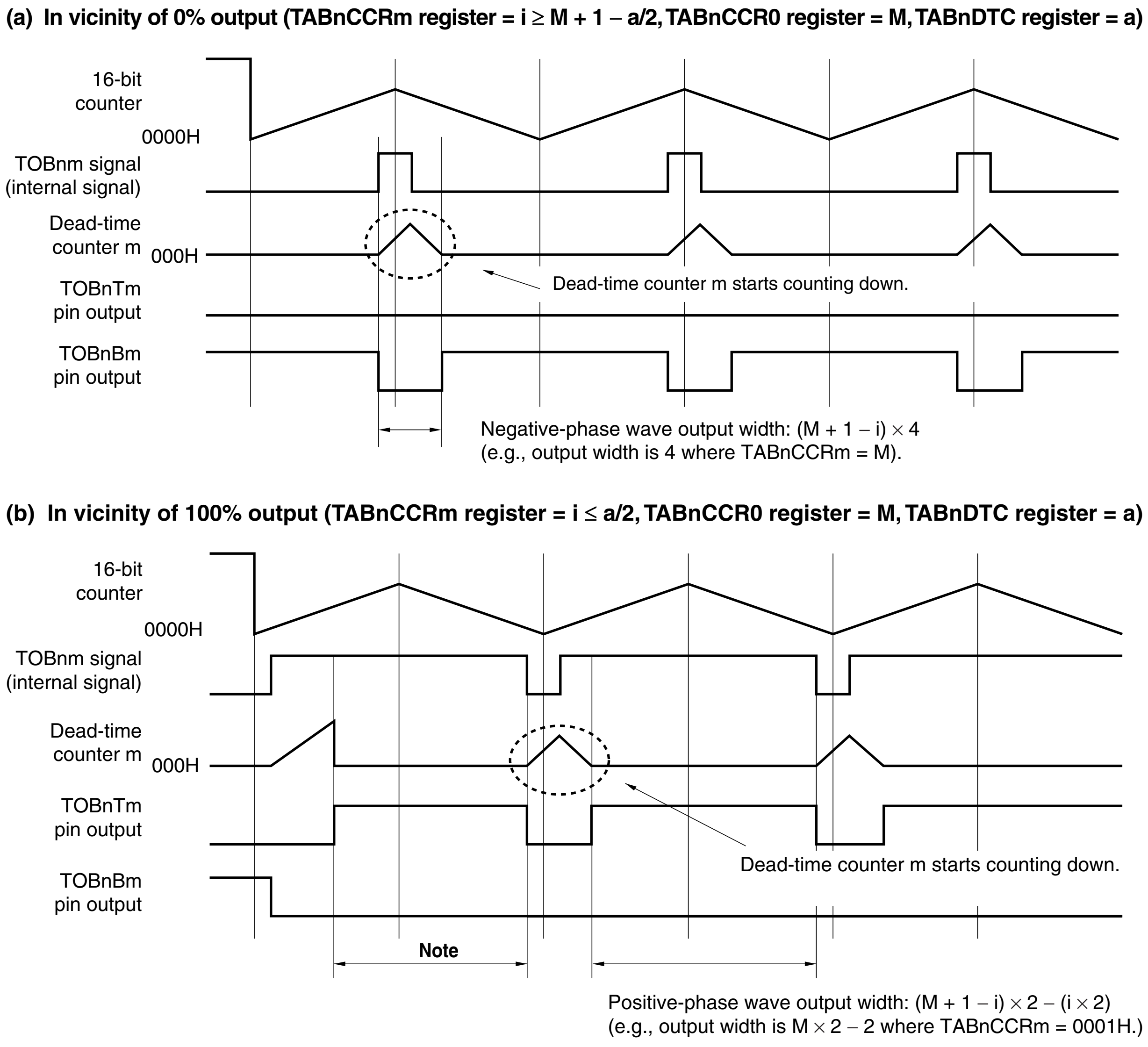
(4) **Automatic dead-time width narrowing function (TABnOPT2.TABnDTM bit = 1)**

The dead-time width can be automatically narrowed in the vicinity of 0% output or 100% output by setting the TABnOPT2.TABnDTM bit to 1.

By setting the TABnDTM bit to 1, the dead-time counter is not cleared, but starts down counting if the TOBnm (internal signal) output of timer AB changes during dead-time counting.

The following timing chart shows the operation of the dead-time counter when the TABnDTM bit is set to 1.

Figure 10-13. Operation of Dead-Time Counter m (1)



(a) In vicinity of 0% output (TABnCCRm register = i ≥ M + 1 − a/2, TABnCCR0 register = M, TABnDTC register = a)

Dead-time counter m starts counting down.

Negative-phase wave output width: (M + 1 − i) × 4
(e.g., output width is 4 where TABnCCRm = M).

(b) In vicinity of 100% output (TABnCCRm register = i ≤ a/2, TABnCCR0 register = M, TABnDTC register = a)

Dead-time counter m starts counting down.

Positive-phase wave output width: (M + 1 − i) × 2 − (i × 2)
(e.g., output width is M × 2 − 2 where TABnCCRm = 0001H.)

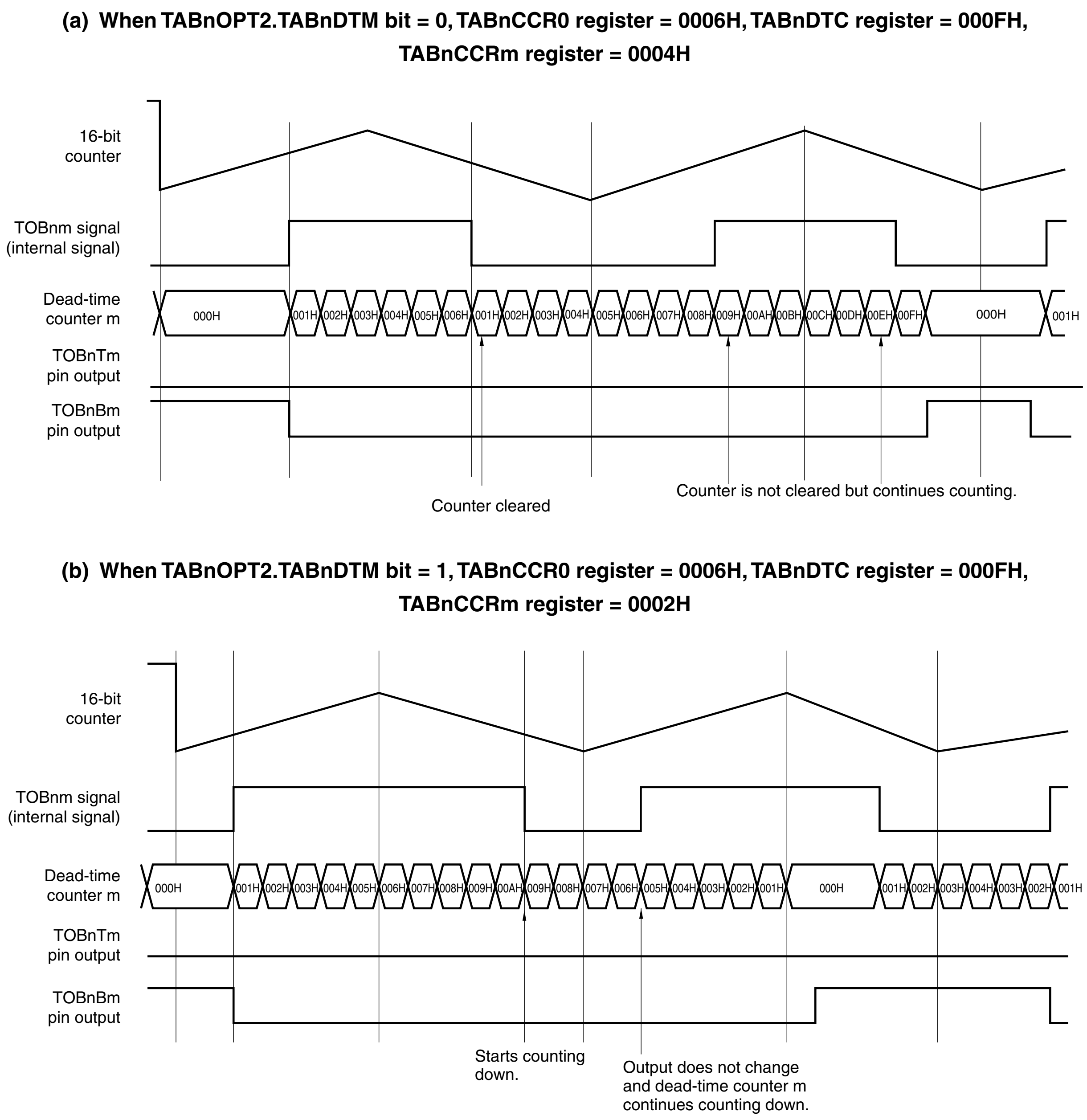
Note The output width of the first wave differs from that of the second and subsequent waves immediately after the TABnCTL0.TABnCE bit has been set. The first wave is shorter than the second wave because the dead time is fully counted.

Remark V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

(5) Dead-time control in case of incorrect setting

Usually, the TOBnm (internal signal) output of TABn changes only once during dead-time counting, only in the vicinity of 0% and 100% output. This section shows an example where the TABnCCR0 register (carrier cycle) and TABnDTC register (dead-time value) are incorrectly set. If these registers are incorrectly set, the TOBnm (internal signal) output of TABn changes more than once during dead-time counting. The following flowchart shows the 6-phase PWM output waveform in this case.

Figure 10-14. Operation of Dead-Time Counter m (2)



(a) When TABnOPT2.TABnDTM bit = 0, TABnCCR0 register = 0006H, TABnDTC register = 000FH, TABnCCRm register = 0004H

16-bit counter
TOBnm signal (internal signal)
Dead-time counter m
000H
001H 002H 003H 004H 005H 006H 001H 002H 003H 004H 005H 006H 007H 008H 009H 00AH 00BH 00CH 00DH 00EH 00FH
000H
001H
TOBnTm pin output
TOBnBm pin output
Counter cleared
Counter is not cleared but continues counting.

(b) When TABnOPT2.TABnDTM bit = 1, TABnCCR0 register = 0006H, TABnDTC register = 000FH, TABnCCRm register = 0002H

16-bit counter
TOBnm signal (internal signal)
Dead-time counter m
000H
001H 002H 003H 004H 005H 006H 007H 008H 009H 00AH 009H 008H 007H 006H 005H 004H 003H 002H 001H
000H
001H 002H 003H 004H 003H 002H 001H
TOBnTm pin output
TOBnBm pin output
Starts counting down.
Output does not change and dead-time counter m continues counting down.

Remark V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

10.4.3 Interrupt culling function

- The interrupts to be culled are INTTBnCC0 (crest interrupt) and INTTBnOV (valley interrupt).
- The TABnOPT1.TABnICE bit is used to enable output of the INTTBnCC0 interrupt and specify the count signal for interrupt culling.
- The TABnOPT1.TABnIOE bit is used to enable output of the INTTBnOV interrupt and specify the count signal for interrupt culling.
- The TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits are used to specify the number of interrupts to be culled, specified for the count signals for interrupt culling.

  The interrupts are masked for the specified number of culling counts and an interrupt occurs at the next interrupt timing.
- The TABnOPT2.TABnRDE bit is used to specify whether transfer is to be culled or not.

  If it is specified that transfer is to be culled, transfer is executed at the same timing as the interrupt output after culling. If it is specified that transfer is not to be culled, transfer is executed at the transfer timing after the TABnCCR1 register has been written.
- The TABnOPT0.TABnCMS bit is used to specify whether the registers with a transfer function are batch rewritten or anytime rewritten.

  The values of the registers are updated in synchronization with transferring when the TABnCMS bit is 0. When the TABnCMS bit is 1, the values of the registers are immediately updated when a new value is written to the registers.

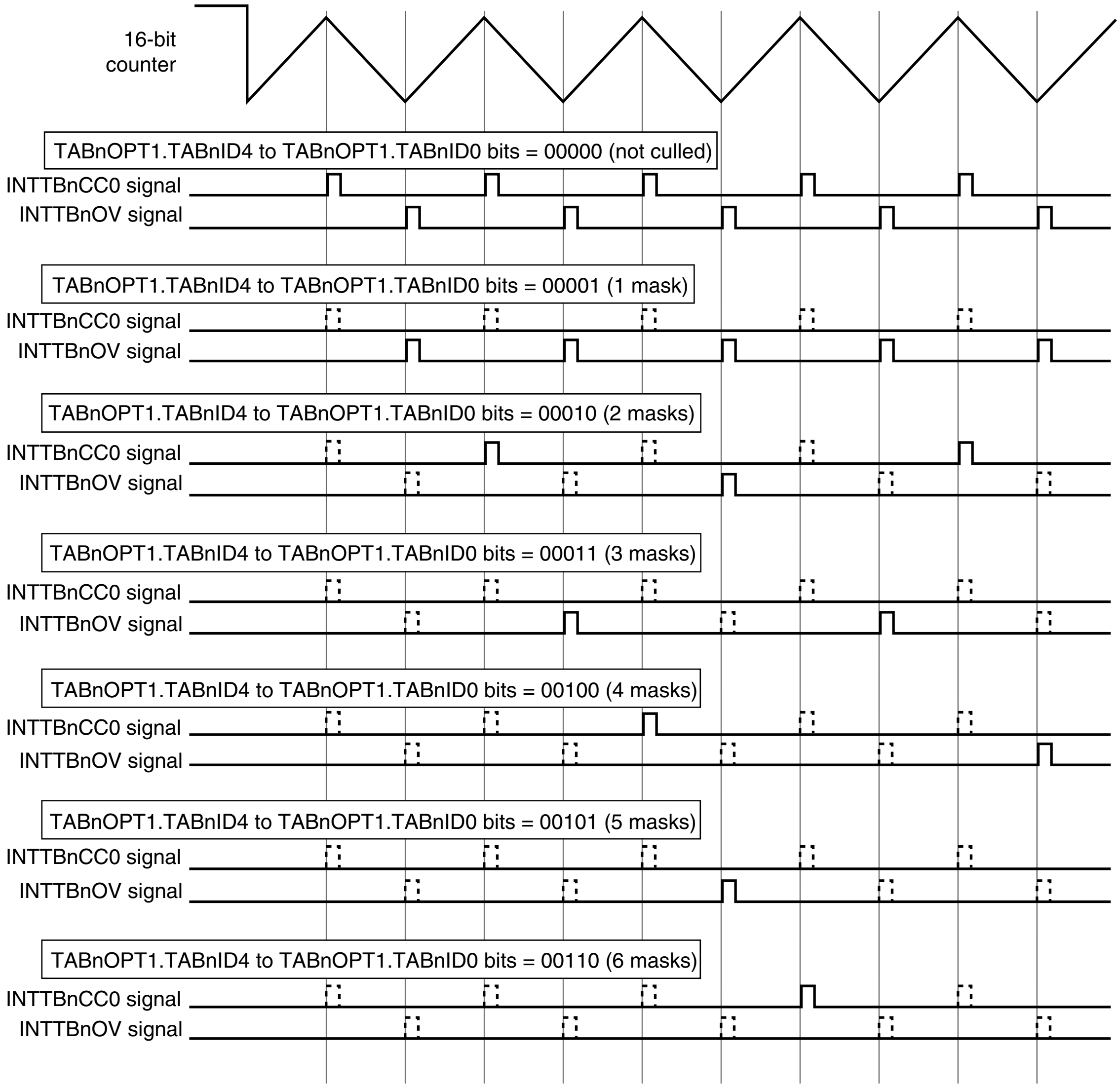
  Transfer is performed from the TABnCCRm register to the CCRm buffer register in synchronization with interrupt culling timing.

Cautions 1. When using the interrupt culling function in the batch rewrite mode (transfer mode), execute the function in the intermittent batch rewrite mode (transfer culling mode).

2. An interrupt is generated at the timing after culling.

(1) Interrupt culling operation

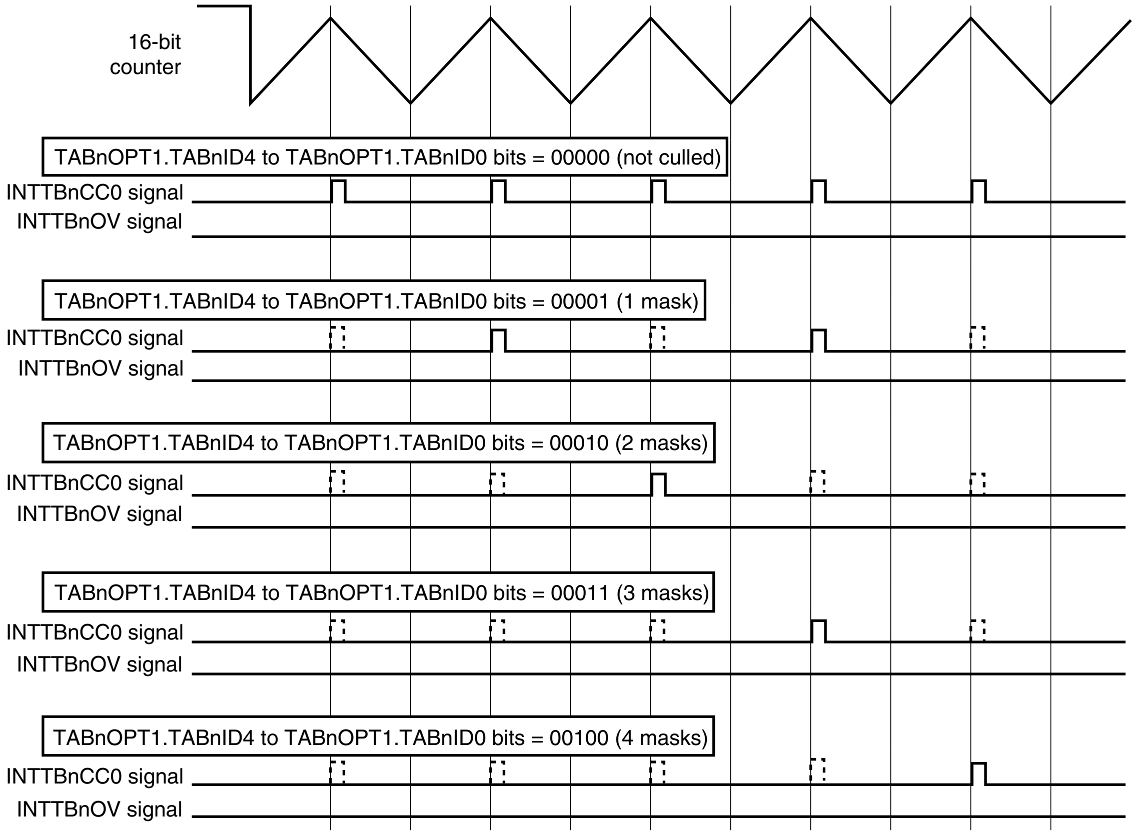
Figure 10-15. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT2.TABnRDE Bit = 1 (Crest/Valley Interrupt Output)


16-bit counter
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00000 (not culled)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001 (1 mask)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00010 (2 masks)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00011 (3 masks)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00100 (4 masks)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00101 (5 masks)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00110 (6 masks)
INTTBnCC0 signal
INTTBnOV signal

Remarks 1. ⌜⌝: Culled interrupt

2. V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

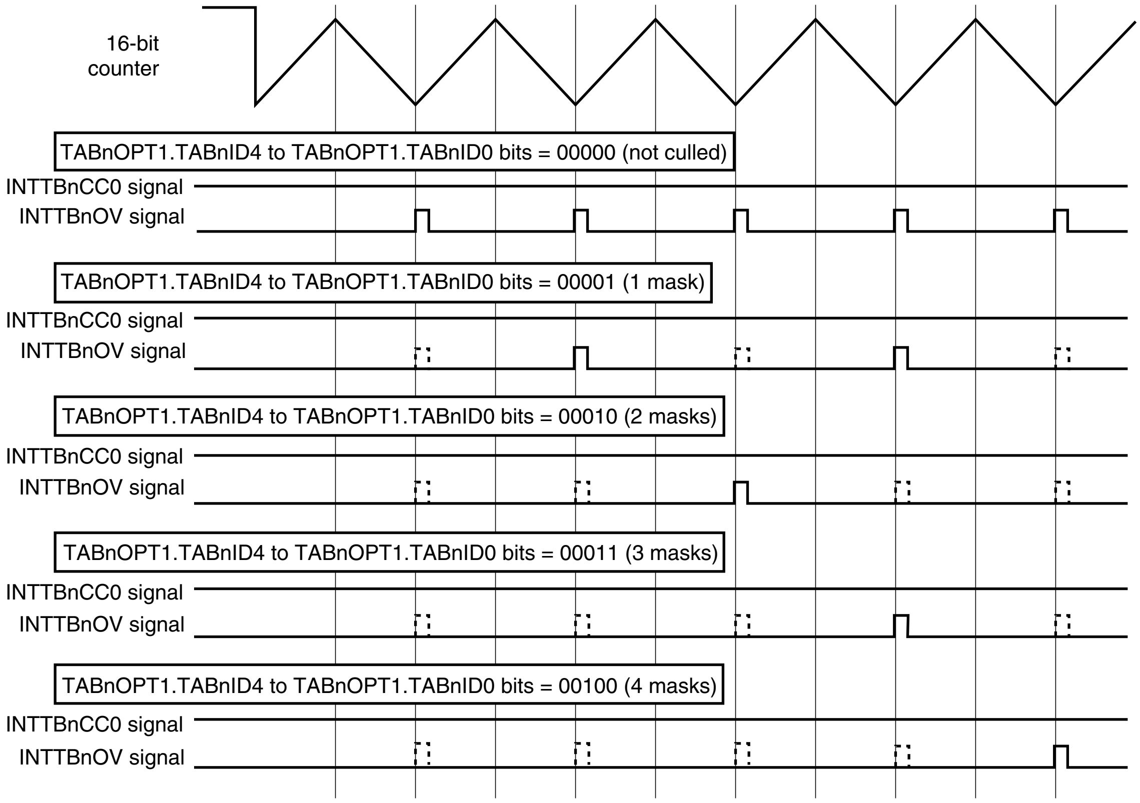
Figure 10-16. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 0, TABnOPT2.TABnRDE Bit = 1 (Crest Interrupt Output)


16-bit counter
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00000 (not culled)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001 (1 mask)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00010 (2 masks)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00011 (3 masks)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00100 (4 masks)
INTTBnCC0 signal
INTTBnOV signal

Remarks 1. ⸽: Culled interrupt

2. V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

Figure 10-17. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT2.TABnRDE Bit = 1 (Valley Interrupt Output)


16-bit counter
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00000 (not culled)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001 (1 mask)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00010 (2 masks)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00011 (3 masks)
INTTBnCC0 signal
INTTBnOV signal
TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00100 (4 masks)
INTTBnCC0 signal
INTTBnOV signal

Remarks 1. ⌐¦: Culled interrupt

2. V850E/IG4-H: n = 0

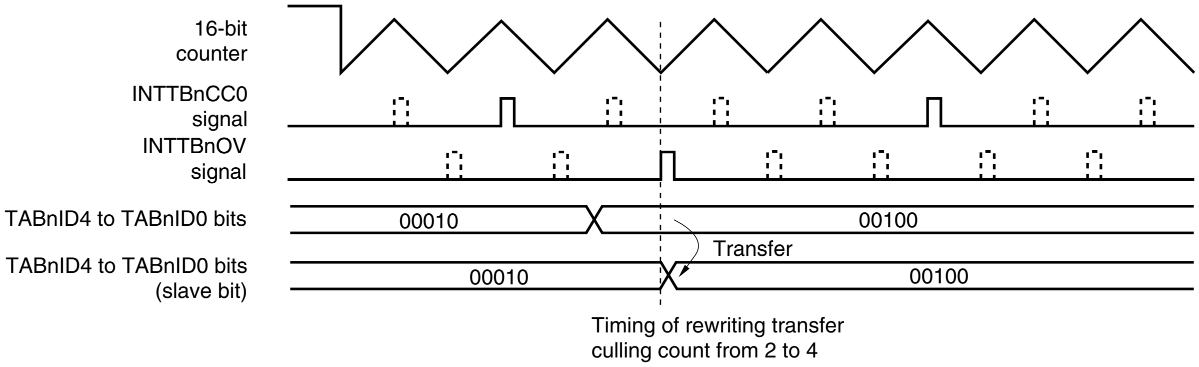
V850E/IH4-H: n = 0, 1

(2) To alternately output crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV)

To alternately output the crest and valley interrupts, set both the TABnOPT1.TABnICE and TABnOPT1.TABnIOE bits to 1.

Figure 10-18. Crest/Valley Interrupt Output

(a) TABnOPT0.TABnCMS bit = 0, TABnOPT2.TABnRDE bit = 1 (with transfer culling control)



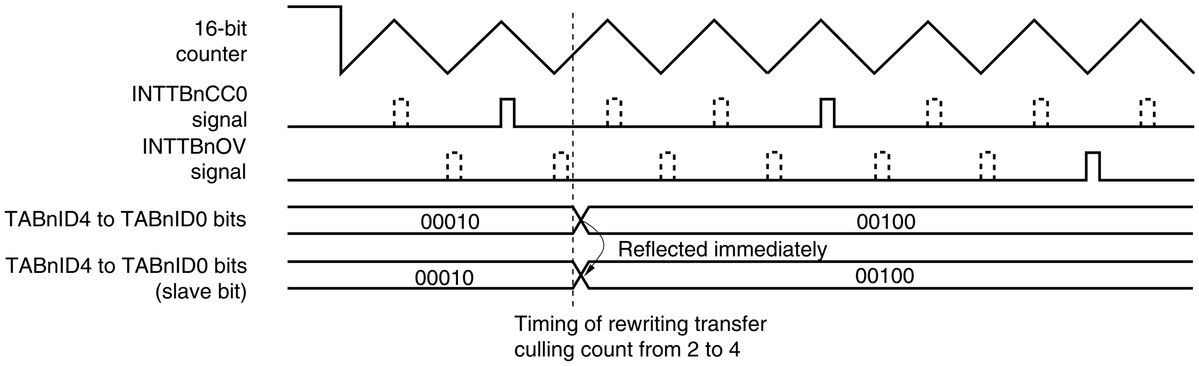
Remarks 1. Transfer is performed when the culled interrupt is output. The other transfer timing is ignored.

2. ⎍: Culled interrupt

3. V850E/IG4-H: n = 0

V850E/IH4-H: n = 0, 1

(b) TABnCMS bit = 1, TABnRDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. ⎍: Culled interrupt

3. V850E/IG4-H: n = 0

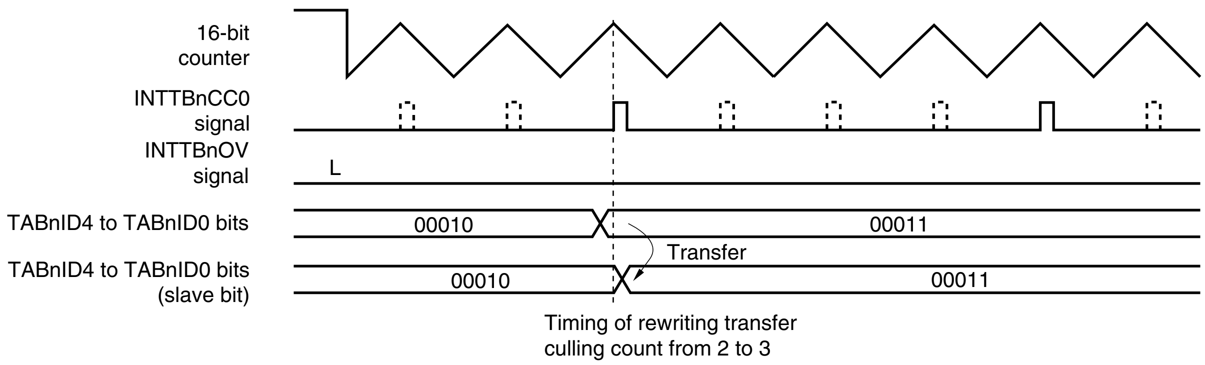
V850E/IH4-H: n = 0, 1

(3) To output only crest interrupt (INTTBnCC0)

Set the TABnOPT1.TABnICE bit to 1 and set the TABnOPT1.TABnIOE bit to 0.

Figure 10-19. Crest Interrupt Output

(a) TABnOPT0.TABnCMS bit = 0, TABnOPT2.TABnRDE bit = 1 (with transfer culling control)



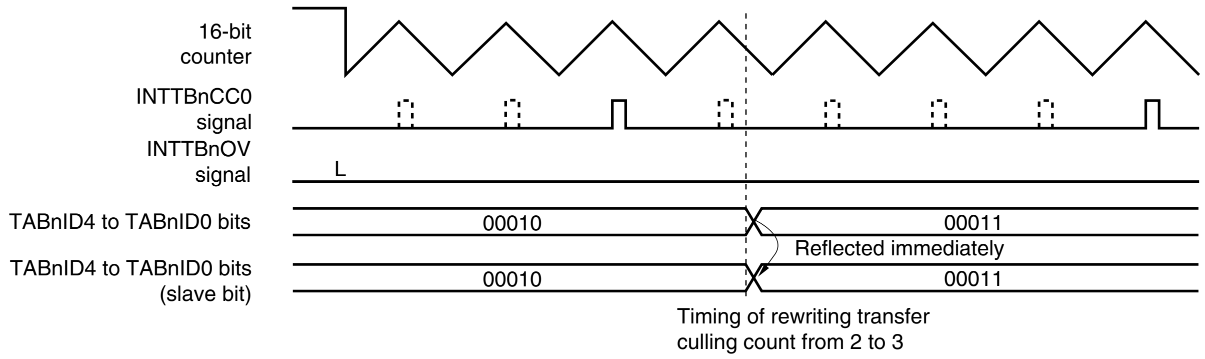
Remarks 1. Transfer is performed when the culled interrupt is output. The other transfer timing is ignored.

2. ¦‾¦: Culled interrupt

3. V850E/IG4-H: n = 0

V850E/IH4-H: n = 0, 1

(b) TABnOPT0.TABnCMS bit = 1, TABnOPT0.TABnRDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. ¦‾¦: Culled interrupt

3. V850E/IG4-H: n = 0

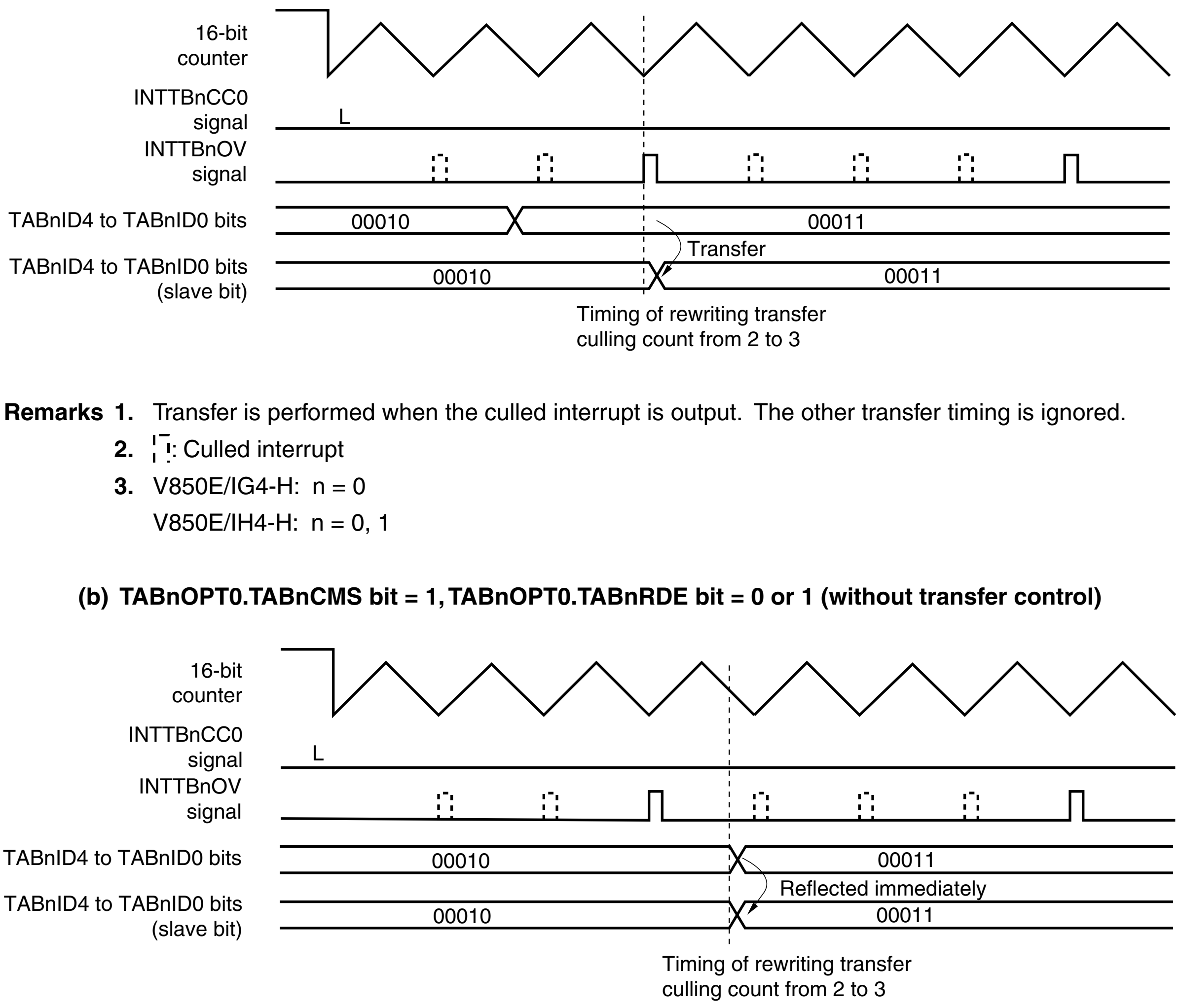
V850E/IH4-H: n = 0, 1

(4) To output only valley interrupt (INTTBnOV)

Set the TABnOPT1.TABnICE bit to 0 and set the TABnOPT1.TABnIOE bit to 1.

Figure 10-20. Valley Interrupt Output

(a) TABnOPT0.TABnCMS bit = 0, TABnOPT2.TABnRDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed when the culled interrupt is output. The other transfer timing is ignored.

2. ⊓: Culled interrupt

3. V850E/IG4-H: n = 0

V850E/IH4-H: n = 0, 1

(b) TABnOPT0.TABnCMS bit = 1, TABnOPT0.TABnRDE bit = 0 or 1 (without transfer control)

Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. ⊓: Culled interrupt

3. V850E/IG4-H: n = 0

V850E/IH4-H: n = 0, 1

10.4.4 Operation to rewrite register with transfer function

The following seven registers are provided with a transfer function and used to control a motor. Each of registers has a buffer register.

- TABnCCR0: Register that specifies the cycle of the 16-bit counter (TAB)
- TABnCCR1: Register that specifies the duty factor of TOBnT1 (U) and TOBnB1 ($\overline{\text{U}}$)
- TABnCCR2: Register that specifies the duty factor of TOBnT2 (V) and TOBnB2 ($\overline{\text{V}}$)
- TABnCCR3: Register that specifies the duty factor of TOBnT3 (W) and TOBnB3 ($\overline{\text{W}}$)
- TABnOPT1: Register that specifies the culling of interrupts
- TAAnCCR0: Register that specifies the A/D conversion start trigger generation timing (TAAn during tuning operation)
- TAAnCCR1: Register that specifies the A/D conversion start trigger generation timing (TAAn during tuning operation)

The following three rewrite modes are provided in the registers with a transfer function.

- Anytime rewriting mode

  This mode is specified by setting the TABnOPT0.TABnCMS bit to 1. The setting of the TABnOPT2.TABnRDE bit is ignored.

  In this mode, each compare register is updated independently, and the value of the compare register is updated as soon as a new value is written to it.

- Batch rewrite mode (transfer mode)

  This mode is specified by setting the TABnOPT0.TABnCMS bit to 0, the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to 00000, and the TABnOPT2.TABnRDE bit to 0.

  When data is written to the TABnCCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TABnCCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.

  The transfer timing is the timing of each crest (match between the 16-bit counter value and TABnCCR0 register value) and valley (match between the 16-bit counter value and 0001H) regardless of the interrupt.

- Intermittent batch rewrite mode (transfer culling mode)

  This mode is specified by setting the TABnOPT0.TABnCMS bit to 0 and the TABnOPT2.TABnRDE bit to 1.

  When data is written to the TABnCCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TABnCCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.

  If interrupt culling is specified by the TABnOPT1 register, the transfer timing is also culled as the interrupts are culled, and the seven registers are transferred all at once at the culled timing of crest interrupt (match between the 16-bit counter value and TABnCCR0 register value) or valley interrupt (match between the 16-bit counter value and 0001H).

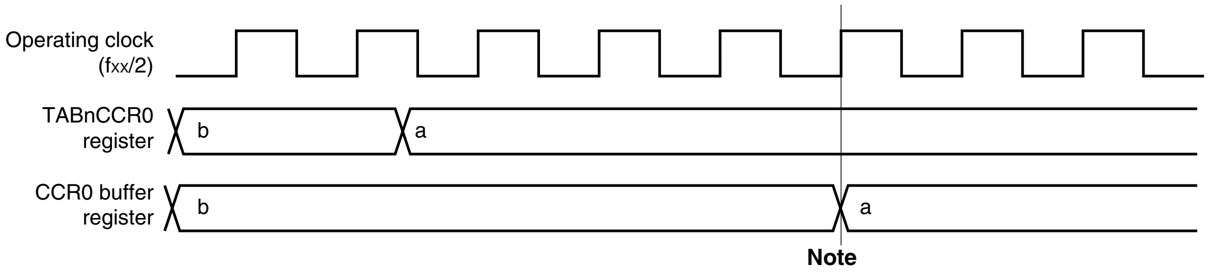
  For details of the interrupt culling function, see **10.4.3 Interrupt culling function**.

(1) Anytime rewriting mode

This mode is specified by setting the TABnOPT0.TABnCMS bit is 1. The setting of the TABnOPT2.TABnRDE bit is ignored.

In this mode, the value written to each register with a transfer function is immediately transferred to an internal buffer register and compared with the count value. If a register with transfer function is rewritten in this mode after the count value of the 16-bit counter matches the value of the TABnCCRm register, the rewritten value is not reflected because the next match is ignored after the first match has occurred. If the register is rewritten during up counting, the new register value becomes valid after the counter has started counting down.

Figure 10-21. Timing of Reflecting Rewritten Value

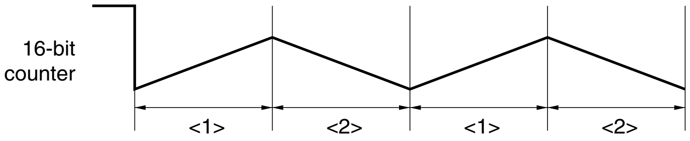


Note After writing to a register (TABnCCR0, TABnCCR2, TABnCCR3, TABnOPT1, TAAnCCR0, or TAAnCCR1), the value is transferred to the internal buffer register during the fourth cycle of the operating clock. However, the value of only the TABnCCR1 register is transferred after 5 more clocks.

(a) Rewriting TABnCCR0 register

Even if the TABnCCR0 register is rewritten in the anytime rewriting mode, the new value may not be reflected in some cases.

Figure 10-22. Example of Rewriting TABnCCR0 Register



Rewriting during period <1> (rewriting during up counting)

If the newly rewritten value is greater than the value of the 16-bit counter, there is no problem because it will match the value of the 16-bit counter. If the new value is less than the value of the 16-bit counter, it will not match the value of the counter. As a result, the 16-bit counter overflows and continues counting up from 0000H until it matches the register value again, and the correct PWM waveform is not output.

Rewriting during period <2> (rewriting during down counting)

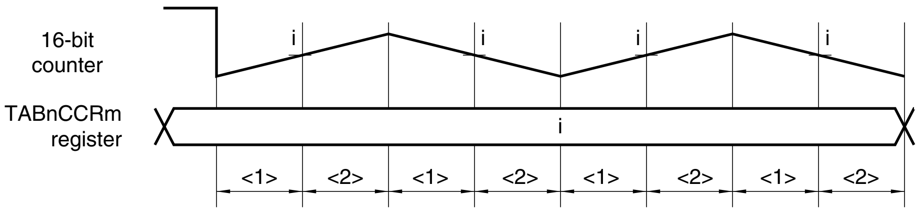
A match with the value of the 16-bit counter is ignored during counting down. Therefore, the rewritten period value is reflected starting from counting up in the next cycle as a match point.

(b) Rewriting TABnCCRm register

Figure 10-24 shows the timing of rewriting before the value of the 16-bit counter matches the value of the TABnCCRm register (<1> in Figure 10-23), and Figure 10-25 shows the timing of rewriting after the value of the 16-bit counter matches the value of the TABnCCRm register (<2> in Figure 10-23).

Figure 10-23. Basic Operation of 16-bit Counter and TABnCCRm Register

(a) Basic figure



Remarks 1. i = Set value of TABnCCRm register

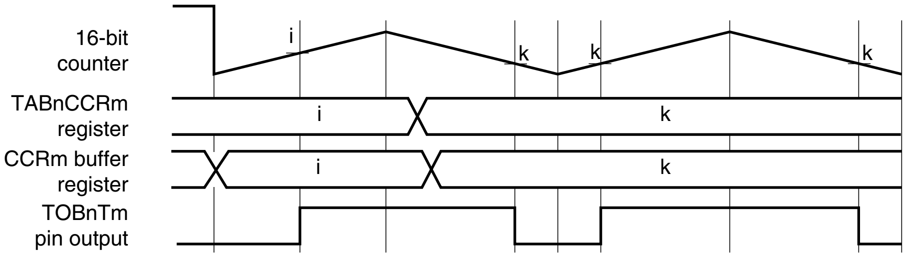
2. V850E/IG4-H: n = 0, m = 1 to 3

V850E/IH4-H: n = 0, 1, m = 1 to 3

Figure 10-24. Example of Rewriting TABnCCR1 to TABnCCR3 Registers (Rewriting Before Match Occurs)

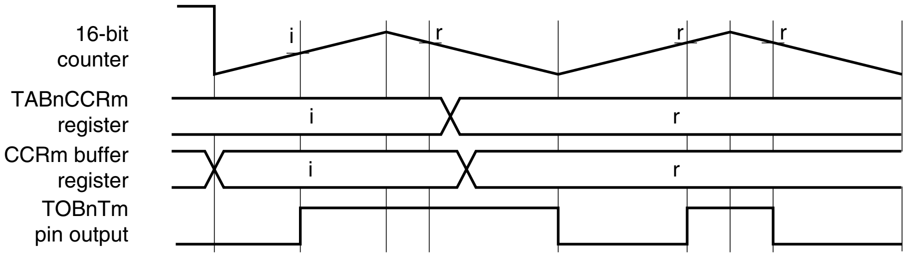
(a)

If the TABnCCRm register is rewritten before its value matches the value of the 16-bit counter, the register value will match the value of the 16-bit counter after the register has been rewritten. Consequently, the new register value is immediately reflected.



(b)

If a value less than the value of the 16-bit counter (greater if the counter is counting down) is written to the TABnCCRm register, the output waveform is as follows because the register value does not match the counter value.

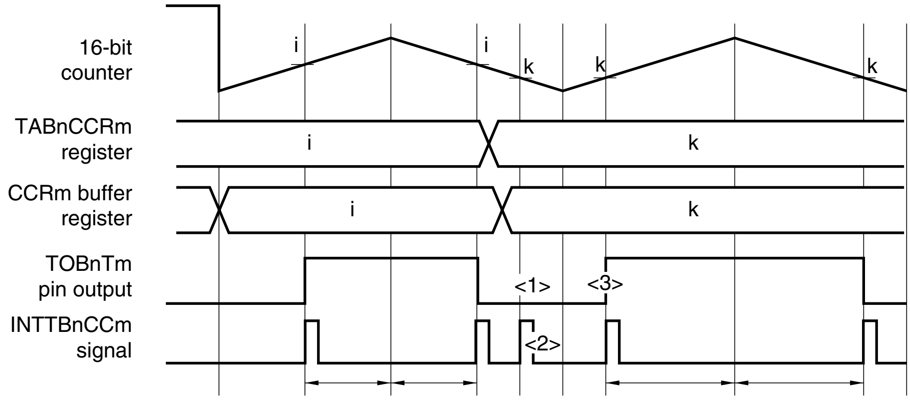


If the register value does not match the counter value, the TOBnTm pin output does not change. Even if the value of the 16-bit counter does not match the value of the TABnCCRm register, the TOBnTm pin output always changes to the high level if the crest interrupt occurs and to the low level if the valley interrupt occurs.
This is a function provided for 0% output and 100% output.
For details, see **10.4.2 (2) PWM output of 0%/100%**.

Remarks 1. i, r, k = Set values of TABnCCRm register

2. V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

Figure 10-25. Example of Rewriting TABnCCR1 to TABnCCR3 Registers (Rewriting After Match Occurs)



<1> Matching of the count value of the 16-bit counter and the value of the TABnCCRm register as a result of rewriting the register is ignored after a match signal has been generated, and the PWM output does not change.

<2> Even if the PWM output does not change, the interrupt generated upon a match between the 16-bit counter value and the TABnCCRm register value (INTTBnCCm) is output.

<3> The next match between the 16-bit counter and TABnCCRm register is valid after the counter has changed its counting direction to up or down, and the PWM output changes.

If the TABnCCRm register is rewritten after its value matches the value of the 16-bit counter, the next match is ignored after the first match occurs and the rewritten value is not reflected to the TOBnTm pin output. If the register is rewritten while the counter is counting up, the match that occurs after the counter starts counting down is valid (the match that occurs after the counter has started counting up is valid if the register is rewritten while the counter is counting down).

Remarks 1. i, r, k = Set value of TABnCCRm register

2. V850E/IG4-H: n = 0, m = 1 to 3
V850E/IH4-H: n = 0, 1, m = 1 to 3

(c) Rewriting TABnOPT1 register

The interrupt culling counter is cleared when the TABnOPT1 register is written. When the interrupt culling counter has been cleared, the measured number of times the interrupt has occurred is discarded. Consequently, the interrupt generation interval is temporarily extended.

To avoid this operation, rewrite the TABnOPT1 register in the intermittent batch rewriting mode (transfer culling mode).

For details of rewriting the TABnOPT1 register, see **10.4.3 Interrupt culling function**.

(2) Batch rewrite mode (transfer mode)

This mode is specified by setting the TABnOPT0.TABnCMS bit to 0, the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to 00000, and the TABnOPT2.TABnRDE bit to 0.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once at the transfer timing and compared with the count value.

(a) Rewriting procedure

If data is written to the TABnCCR1 register, the values set to the TABnCCR0 to TABnCCR3, TABnOPT1, TAAnCCR0, and TAAnCCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TABnCCR1 register last. Writing to the register is prohibited after the TABnCCR1 register has been written and before the transfer timing is generated (until the crest (match between the 16-bit counter value and TABnCCR0 register value) or the valley (match between the 16-bit counter value and 0001H)). The operation procedure is as follows.

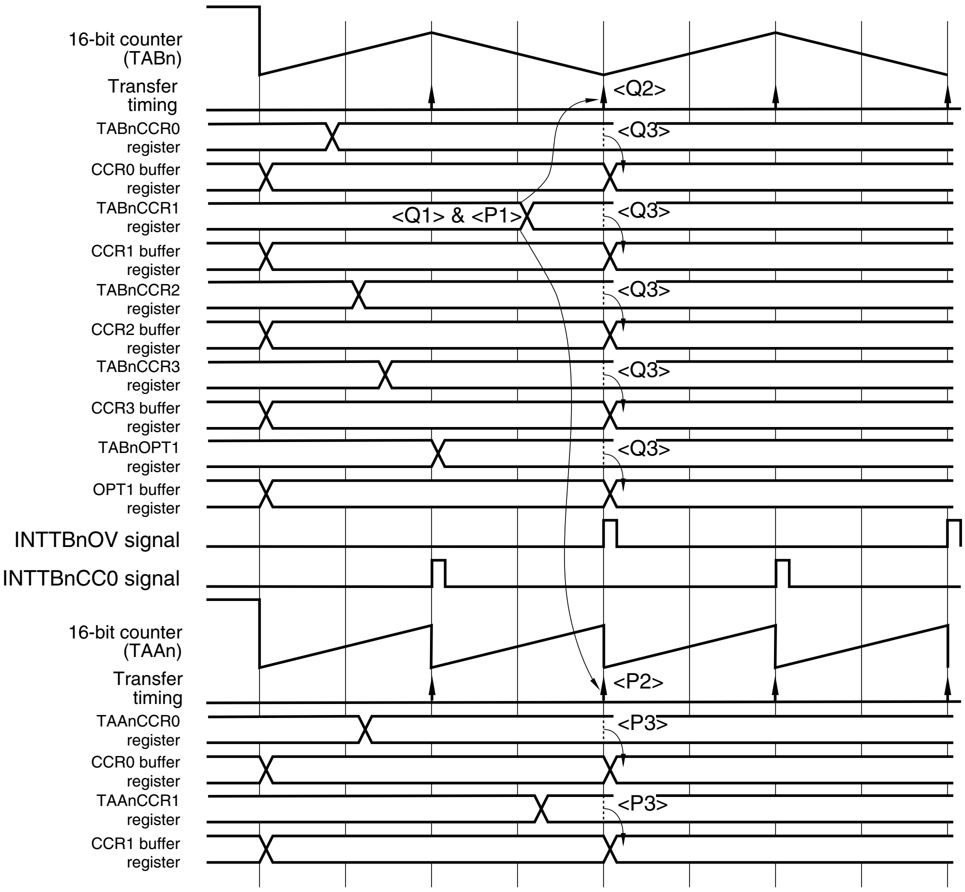
<1> Rewriting the TABnCCR0, TABnCCR2, TABnCCR3, TABnOPT1, TAAnCCR0, and TAAnCCR1 registers.
Do not rewrite registers that do not have to be rewritten.

<2> Rewriting the TABnCCR1 register.
Rewrite the same value to the register even when it is not necessary to rewrite the TABnCCR1 register.

<3> Holding the next rewriting pending until the transfer timing is generated.
Rewrite the register next time after the INTTBnOV or INTTBnCC0 interrupt has occurred.

<4> Return to <1>.

Figure 10-26. Basic Operation in Batch Mode



[Operation of TABn]

<Q1> Write the TABnCCR1 register

<Q2> The target timing is the first transfer timing after a write to the TABnCCR1 register.

<Q3> The values are transferred all at once at the transfer timing.

[Operation of TAAn]

<P1> Write the TABnCCR1 register

<P2> The target timing is the first transfer timing after a write to the TABnCCR1 register.

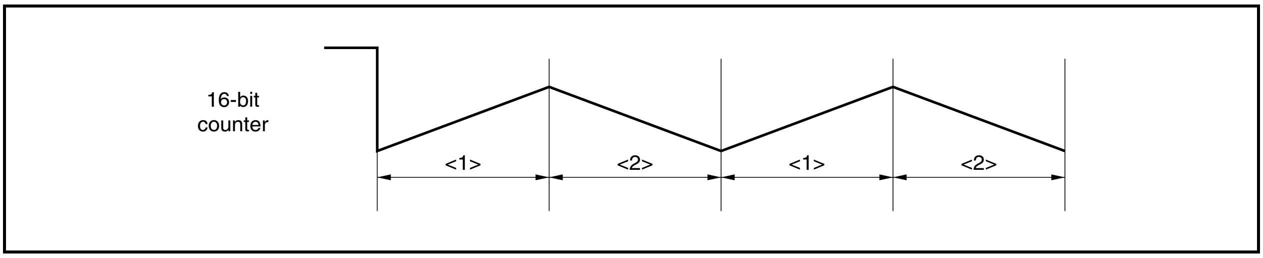
<P3> The values are transferred all at once at the transfer timing.

(b) Rewriting TABnCCR0 register

When rewriting the TABnCCR0 register in the batch rewrite mode, the output waveform differs depending on whether transfer occurs at the crest (match between the 16-bit counter value and TABnCCR0 register value) or at the valley (match between the 16-bit counter value and 0001H). Usually, it is recommended to rewrite the TABnCCR0 register while the 16-bit counter is counting down, and transfer the register value at the transfer timing of the crest timing.

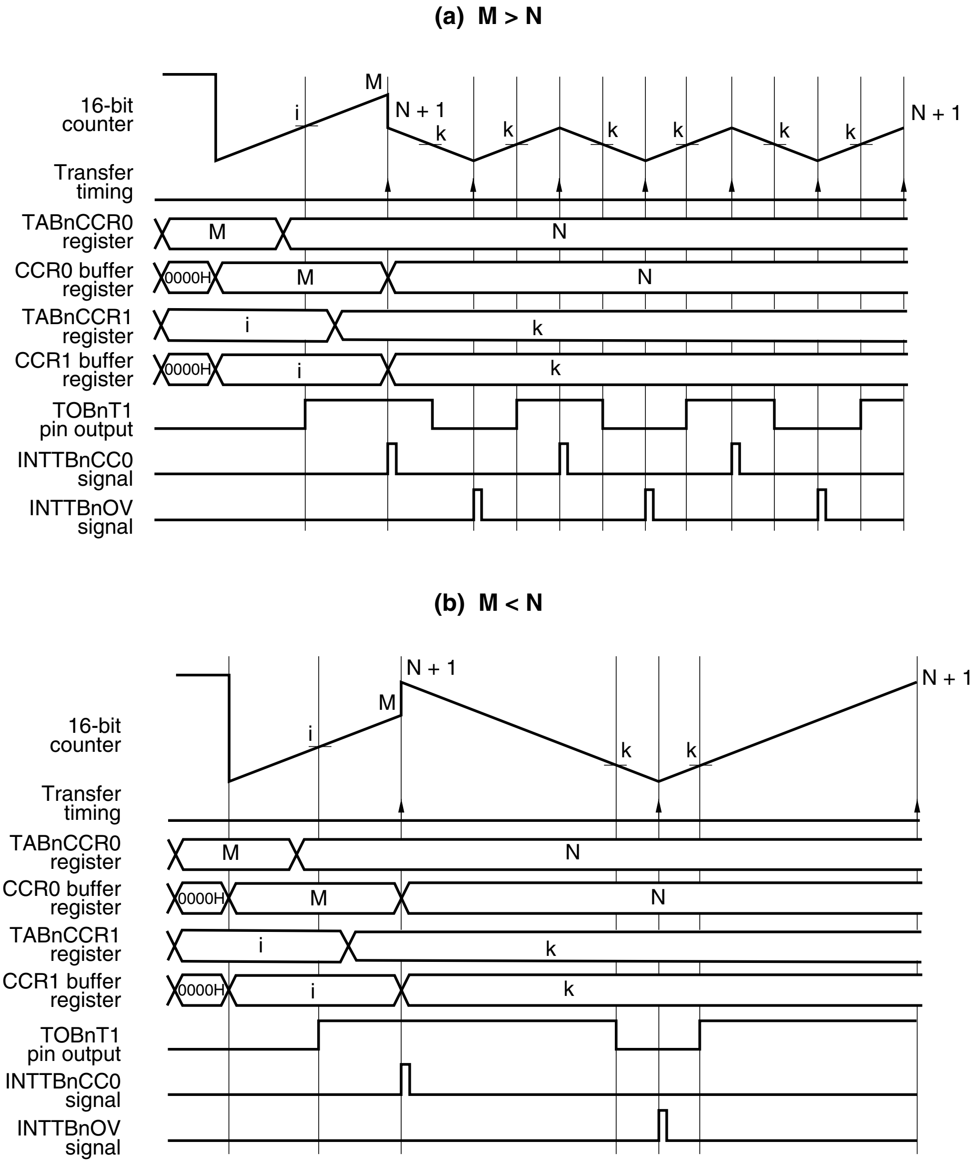
Figure 10-28 shows an example of rewriting the TABnCCR0 register while the 16-bit counter is counting up (during period <1> in Figure 10-27). Figure 10-29 shows an example of rewriting the TABnCCR0 register while the counter is counting down (during period <2> in Figure 10-27).

Figure 10-27. Basic Operation of 16-bit Counter



The transfer timing in Figure 10-28 is at the point where the crest timing occurs. While the 16-bit counter is counting down, the cycle changes and an asymmetrical triangular wave is output. Because the cycle changes, rewrite the duty factor (voltage data value).

Figure 10-28. Example of Rewriting TABnCCR0 Register (During Up Counting)

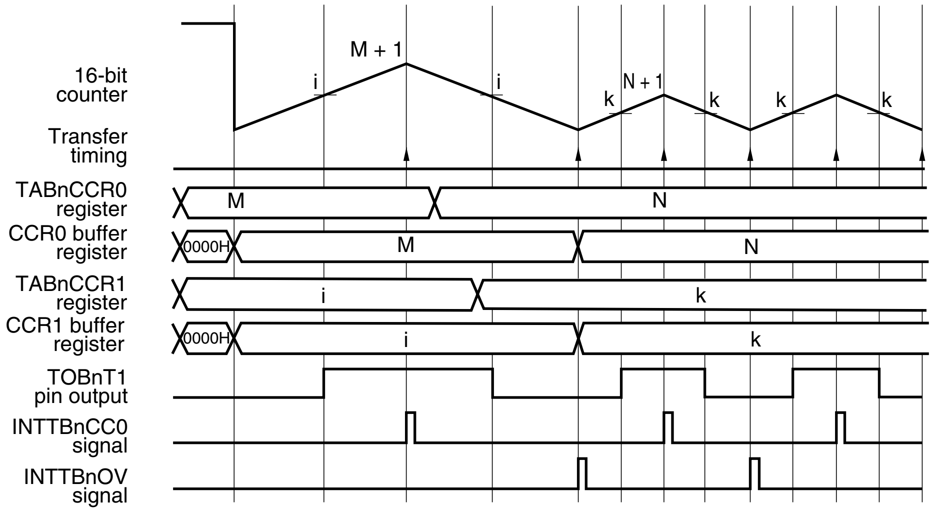


Remarks 1. If transfer (match between the value of the 16-bit counter and the value of the CCR0 buffer register) occurs in the 6-phase PWM output mode, the value of the TABnCCR0 register plus 1 is loaded to the 16-bit counter. In this way, the expected wave can be output even if the cycle value is changed at the transfer timing of the crest (match between the 16-bit counter value and the TABnCCR0 register value) timing.

2. M: Value of CCR0 buffer register before rewriting

N: Value of CCR0 buffer register after rewriting

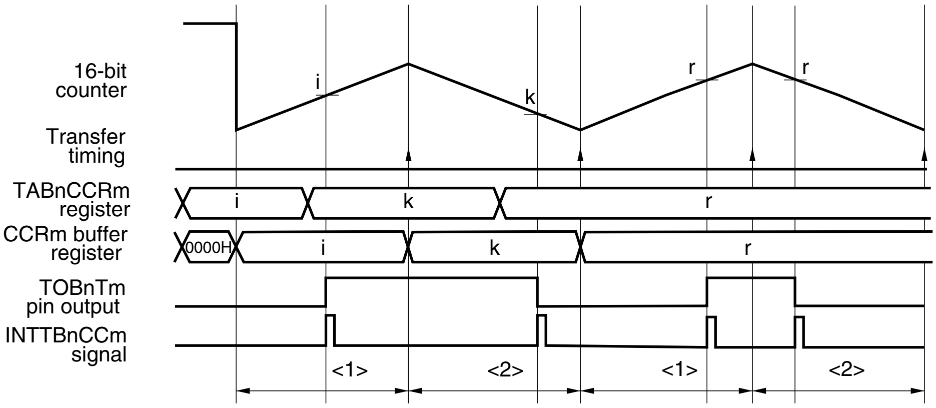
Figure 10-29. Example of Rewriting TABnCCR0 Register (During Down Counting)



Because the next transfer timing is at the point of the valley (match between the 16-bit counter value and 0001H), the cycle value changes from the next cycle and output of a symmetrical triangular wave is maintained. Because the cycle changes, rewrite the duty value (voltage data value) as required.

(c) Rewriting TABnCCRm register

Figure 10-30. Example of Rewriting TABnCCRm Register



Rewriting during period <1> (rewriting during counting up)

Because the TABnCCRm register value is transferred at the transfer timing of the crest (match between the 16-bit counter value and TABnCCR0 register value), an asymmetrical triangular wave is output.

Rewriting during period <2> (rewriting during counting down)

Because the TABnCCRm register value is transferred at the transfer timing of the valley (match between the 16-bit counter value and 0001H), a symmetrical triangular wave is output.

Remark m = 1 to 3

(d) Transferring TABnOPT1 register value

Do not set the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to other than 00000. When using the interrupt culling function, rewrite the TABnOPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TABnOPT1 register, see **10.4.3 Interrupt culling function**.

(3) Intermittent batch rewriting mode (transfer culling mode)

This mode is specified by setting the TABnOPT0.TABnCMS bit is 0 and the TABnOPT2.TABnRDE bit is 1.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once at the culled transfer timing and compared with the count value.

The transfer timing is the timing at which an interrupt is generated (INTTBnCC0, INTTBnOV) by interrupt culling.

For details of the interrupt culling function, see **10.4.3 Interrupt culling function**.

(a) Rewriting procedure

If data is written to the TABnCCR1 register, the TABnCCR0 to TABnCCR3, TABnOPT1, TAAnCCR0, and TAAnCCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TABnCCR1 register last. Writing to the register is prohibited after the TABnCCR1 register has been written until the transfer timing is generated (until the INTTBnOV or INTTBnCC0 interrupt occurs). The operation procedure is as follows.

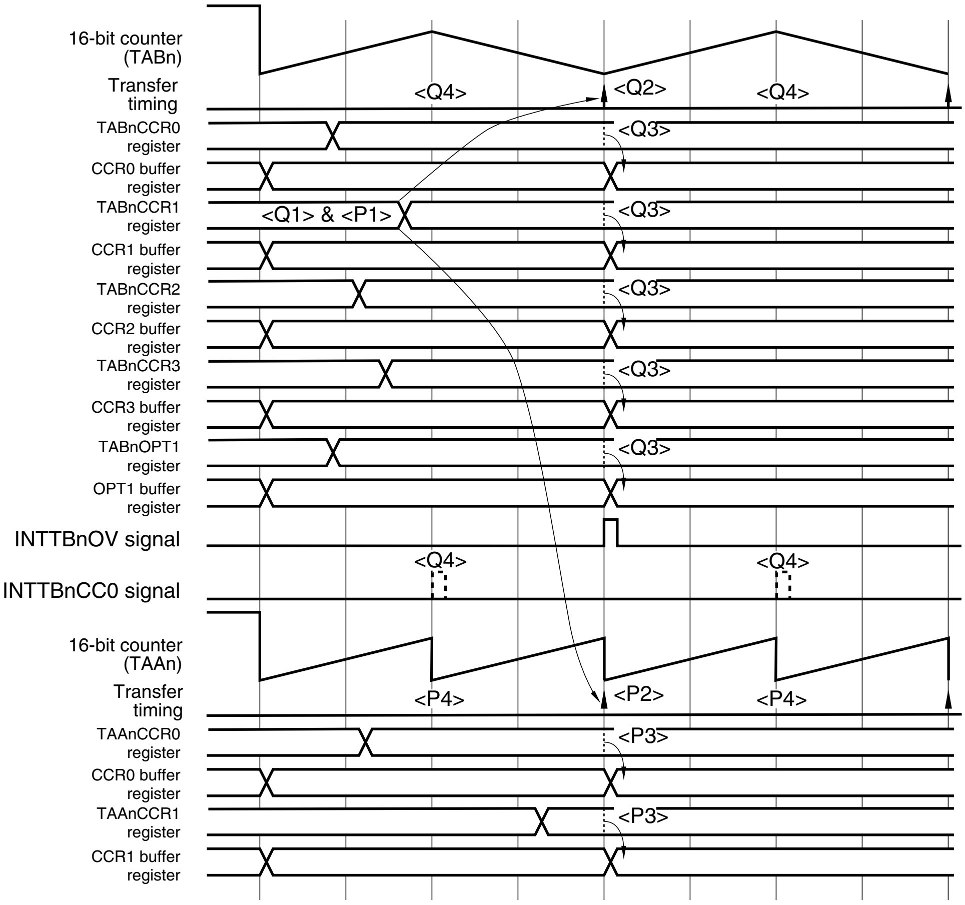
<1> Rewrite the TABnCCR0, TABnCCR2, TABnCCR3, TABnOPT1, TAAnCCR0, and TAAnCCR1 registers.
Do not rewrite registers that do not have to be rewritten.

<2> Rewrite the TABnCCR1 register.
Rewrite the same value to the register even when it is not necessary to rewrite the TABnCCR1 register.

<3> Hold the next rewriting pending until the transfer timing is generated.
Perform the next rewrite after the INTTBnOV or INTTBnCC0 interrupt has occurred.

<4> Return to <1>.

Figure 10-31. Basic Operation in Intermittent Batch Rewriting Mode



[TABn operation]

<Q1> Write the TABnCCR1 register.

<Q2> Rewrite the register at the transfer timing that is generated after the TABnCCR1 register has been rewritten.

<Q3> The registers are transferred all at once at the transfer timing.

<Q4> The transfer timing is also culled as the interrupts are culled.

[TAAn operation]

<P1> Write the TABnCCR1 register.

<P2> Rewrite the register at the transfer timing that is generated after the TABnCCR1 register has been rewritten.

<P3> The registers are transferred all at once at the transfer timing.

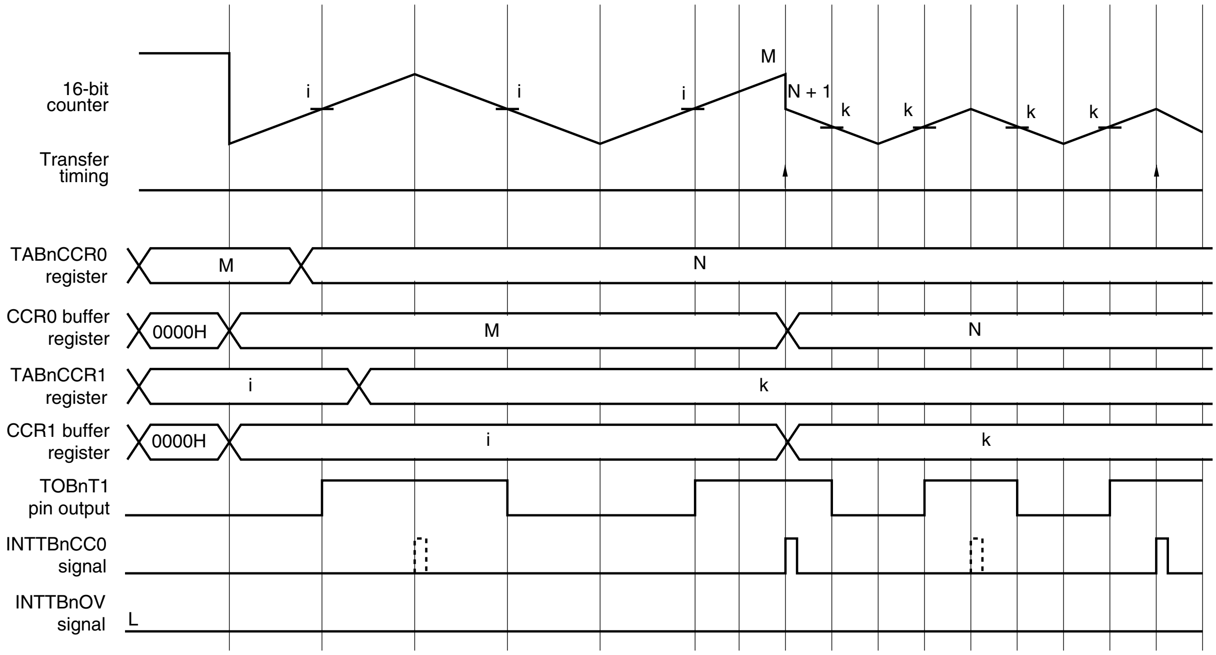
<P4> The transfer timing is also culled as the interrupts are culled.

Remark This is an example of the operation when the TABnOPT1.TABnICE bit = 1, TABnOPT1.TABnIOE bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001.

(b) Rewriting TABnCCR0 register

When rewriting the TABnCCR0 register in the intermittent batch mode, the output waveform differs depending on where the occurrence of the crest or valley interrupt is specified by the interrupt culling setting. The following figure illustrates the change of the output waveform when interrupts are culled.

Figure 10-32. Rewriting TABnCCR0 Register (When Crest Interrupt Is Set)



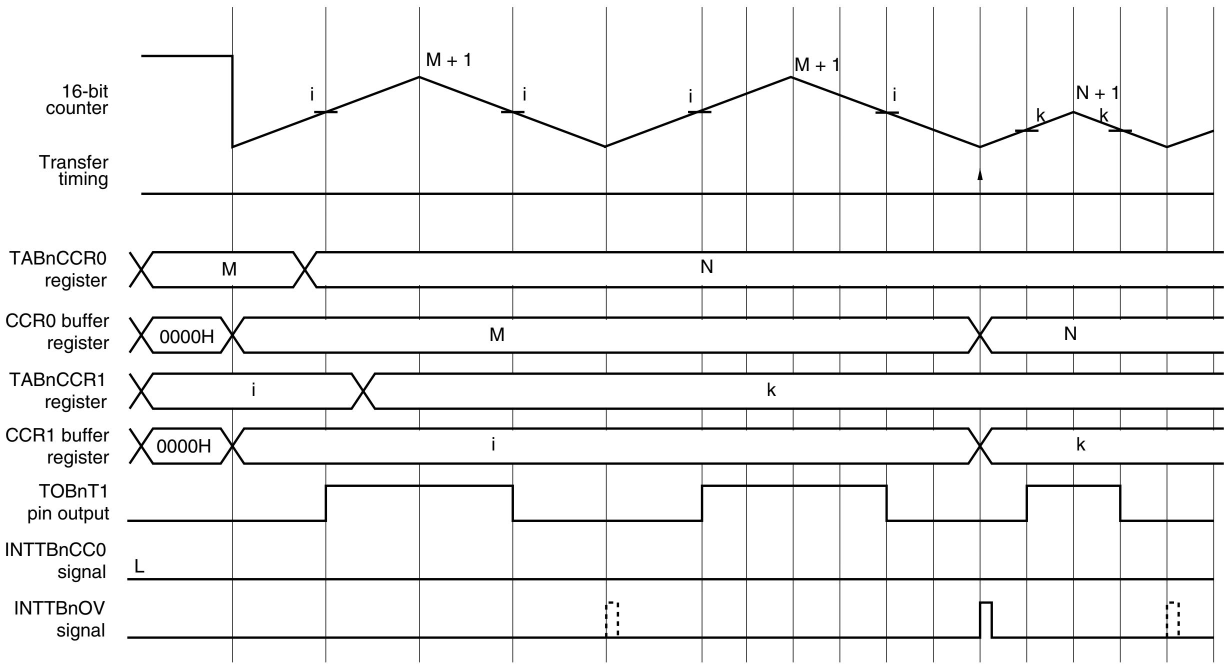
The transfer timing is generated when the crest interrupt occurs, the period of up counting and down counting changes, and an asymmetrical triangular wave is output.

Remarks 1. This is an example of the operation when the TABnOPT1.TABnICE bit = 1, TABnOPT1.TABnIOE bit = 0, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001.

2. ┊┊: Culled interrupt

3. V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

Figure 10-33. Rewriting TABnCCR0 Register (When Valley Interrupt Is Set)



The transfer timing is generated when the valley interrupt occurs, the cycle of up counting and down counting becomes identical, and a symmetrical triangular wave is output.

Remarks 1. This is an example of the operation when the TABnOPT1.TABnICE bit = 0, TABnOPT1.TABnIOE bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001.

2. ╎╎: Culled interrupt

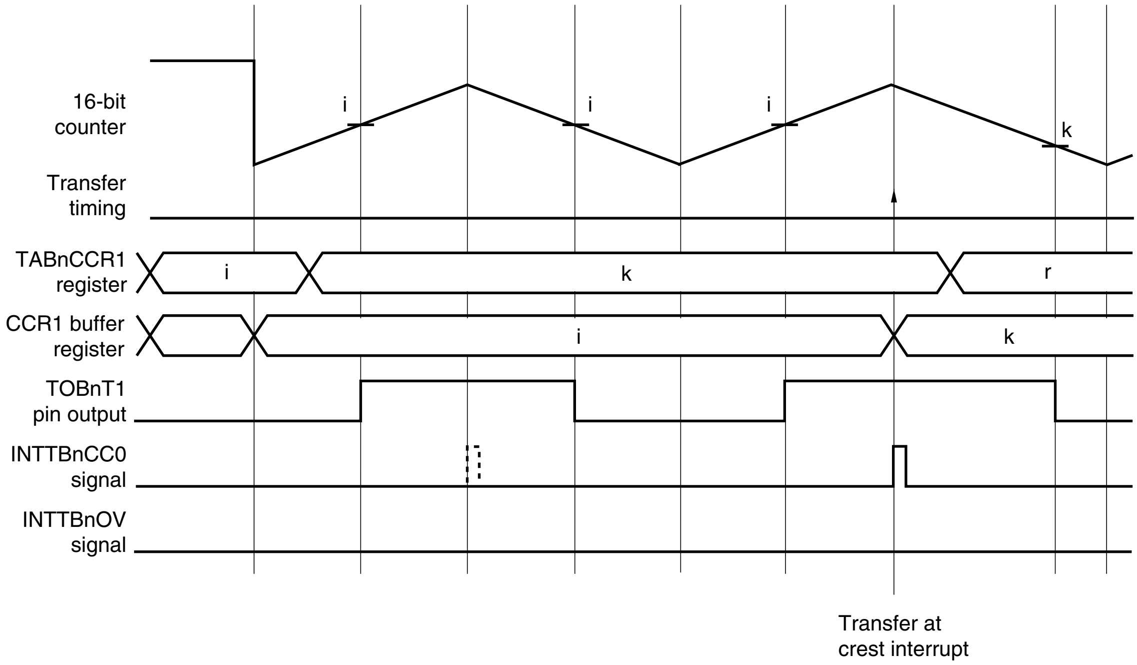
3. V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

(c) Rewriting TABnCCR1 to TABnCCR3 registers

- Transfer at crest when crest interrupt is set

  Because the register is transferred at the transfer timing of the crest interrupt, an asymmetrical triangular wave is output.

Figure 10-34. Rewriting TABnCCR1 Register (TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 0, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 = 00001)



Remarks 1. ┆: Culled interrupt

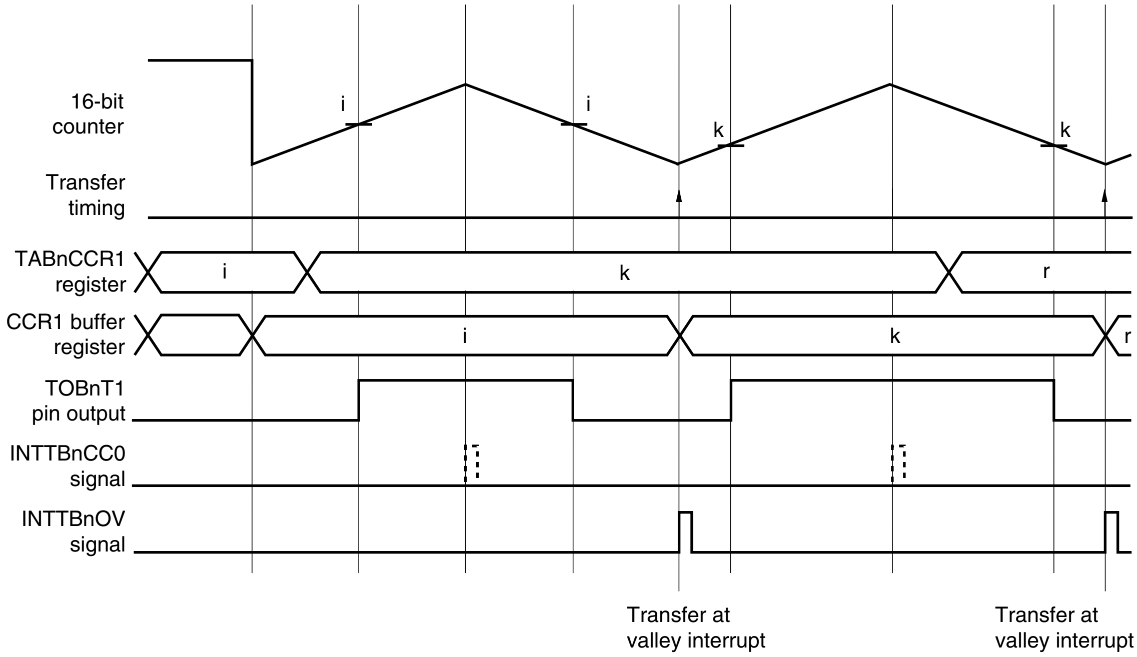
2. V850E/IG4-H: n = 0

V850E/IH4-H: n = 0, 1

- Transfer at valley when valley interrupt is set

  Because the register is transferred at the transfer timing of the valley interrupt, a symmetrical triangular wave is output.

Figure 10-35. Rewriting TABnCCR1 Register
(TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 = 00001)



Remarks 1. ¦‾¦: Culled interrupt

2. V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

(d) Rewriting TABnOPT1 register

Because a new interrupt culling value is transferred when the value of the interrupt culling counter matches the value of the 16-bit counter, the next interrupt and those that follow occur at the set interval. For details of rewriting the TABnOPT1 register, see **10.4.3 Interrupt culling function**.

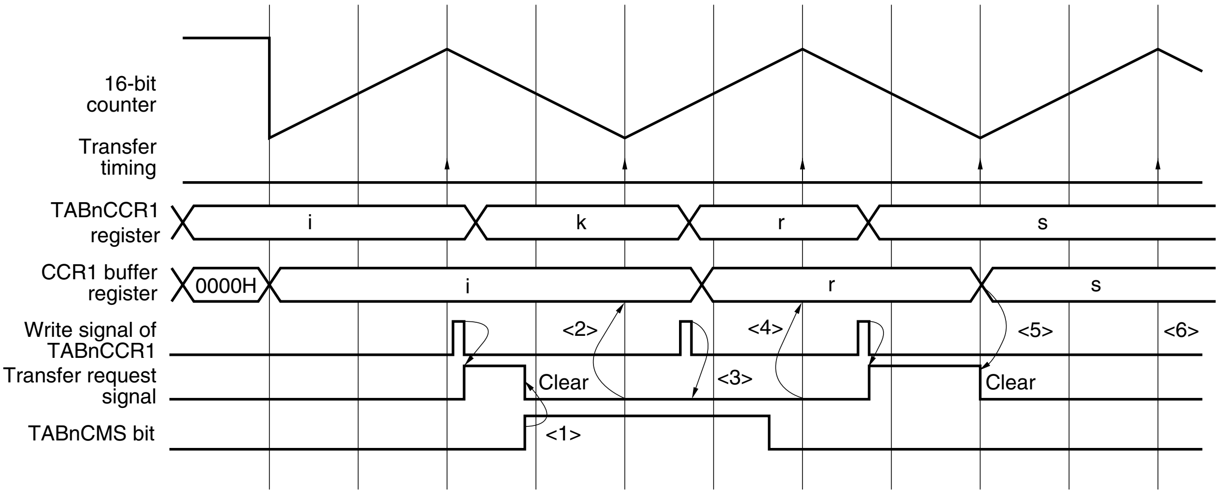
(4) Rewriting TABnOPT0.TABnCMS bit

The TABnCMS bit can select the anytime rewrite mode and batch rewrite mode. This bit can be rewritten during timer operation (when TABnCTL0.TABnCE bit = 1). However, the operation and caution illustrated in Figure 10-36 are necessary.

If the TABnCCR1 register is written when the TABnCMS bit is set to 0, a transfer request signal (internal signal) is set.

When the transfer request signal is set, the register is transferred at the next transfer timing, and the transfer request signal is cleared. This transfer request signal is also cleared when the TABnCMS bit is set to 1.

Figure 10-36. Rewriting TABnCMS Bit



<1> If the TABnCCR1 register is rewritten when the TABnCMS bit is 0, the transfer request signal is set.
If the TABnCMS bit is set to 1 in this status, the transfer request signal is cleared.

<2> The register is not transferred because the TABnCMS bit is set to 1 and the transfer request signal is cleared.

<3> The transfer request signal is not set even if the TABnCCR1 register is written when the TABnCMS bit is 1.

<4> The transfer request signal is not set even if the TABnCCR1 register is written when the TABnCMS bit is 1, so even if the TABnCMS bit is set to 0, transfer does not occur at the subsequent transfer timing.

<5> The transfer request signal is set if the TABnCCR1 register is written when the TABnCMS bit is 0.
Transfer is performed at the subsequent transfer timing and the transfer request signal is cleared.

<6> Once transfer has been performed, the transfer request signal is cleared. Therefore, transfer is not performed at the next transfer timing.

Remark V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

10.4.5 TAAn tuning operation for A/D conversion start trigger signal output

This section explains the tuning operation of TAAn and TABn in the 6-phase PWM output mode.

In the 6-phase PWM output mode, the tuning operation is performed with TABn serving as the master and TAAn as a slave. The conversion start trigger signal of A/D converters 0 and 1 can be set as the A/D conversion start trigger source by the INTTAnCC0 and INTTAnCC1 signals of TAAn and the INTTBnOV and INTTBnCC0 signals of TABn.

Remark V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

(1) Tuning operation starting procedure

The TAAn and TABn registers should be set using the following procedure to perform the tuning operation.

(a) Setting of TAAn register (stop the operations of TABn and TAAn (by setting the TABnCTL0.TABnCE bit and TAAnCTL0.TAAnCE bit to 0))

- Set the TAAnCTL1 register to 85H (set the tuning operation slave mode and free-running timer mode).
- Set the TAAnOPT0 register to 00H (select the compare register).
- Set an appropriate value to the TAAnCCR0 and TAAnCCR1 registers (set the default value for comparison for starting the operation).

(b) Setting of TABn register

- Set the TABnCTL1 register to 07H (set the master mode and 6-phase PWM output mode).
- Set an appropriate value to the TABnIOC0 register (set the output mode of TOBnT1 to TOBnT3). However, set the TABnOL0 bit to 0 and the TABnOE0 bit to 1 (enable positive phase output). Unless this setting is made, the crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV) do not occur. Consequently, the conversion start trigger signal of A/D converters 0 and 1 is not correctly generated.
- Clear the TABnIOC1 and TABnIOC2 registers to 00H (the TIBn0 to TIBn3, EVTBn, and TRGBn pins of TABn are not used).
- Clear the TABnOPT0 register to 00H (select the compare register).
- Set an appropriate value to the TABnCCR0 to TABnCCR3 registers (set the default value for comparison for starting the operation).
- Set the TABnCTL0 register to 0xH (set the TABnCE bit to 0 and the operating clock of TABn). The operating clock of TABn set by the TABnCTL0 register is also supplied to TAAn, and the count operation is performed at the same timing. The operating clock of TAAn set by the TAAnCTL0 register is ignored.

(c) Setting of TMQOPn (TMQn option) register

- Set an appropriate value to the TABnOPT1 and TABnOPT2 registers.
- Set an appropriate value to the TABnIOC3 register (set TOBnB1 to TOBnB3 in the output mode).
- Set an appropriate value to the TABnDTC register (set the default value for comparison for starting the operation).

(d) Setting of alternate function

- Select the alternate function of the port by setting the port to the port control mode.

(e) **Set the TAAnCE bit to 1 and set the TABnCE bit to 1 immediately after that to start the 6-phase PWM output operation.**

Rewriting the TABnCTL0, TABnCTL1, TABnIOC1, TABnIOC2, TAAnCTL0, and TAAnCTL1 registers is prohibited during operation. The operation and the PWM output waveform are not guaranteed if any of these registers is rewritten during operation. However, rewriting the TABnCTL0.TABnCE bit to clear it is permitted. Manipulating (reading/writing) the other TABn, TAAn, and TMQn option registers is prohibited until the TAAnCTL0.TAAnCE bit is set to 1 and then the TABnCE bit is set to 1.

(2) Tuning operation clearing procedure

To clear the tuning operation and exit the 6-phase PWM output mode, set the TAAn and TABn registers using the following procedure.

<1> Clear the TABnCTL0.TABnCE bit to 0 and stop the timer operation.
<2> Clear the TAAnCTL0.TAAnCE bit to 0 so that TAAn can be separated.
<3> Stop the timer output by using the TABnIOC0 register.
<4> Clear the TAAnCTL1.TAAnSYE bit to 0 to clear the tuning operation.

Caution Manipulating (reading/writing) the other TABn, TAAn, and TMQn option registers is prohibited until the TABnCE bit is set to 0 and then the TAAnCE bit is set to 0.

(3) When not tuning TAAn

When the match interrupt signal of TAAn is not necessary as the conversion trigger source that starts A/D converters 0 and 1, TAAn can be used independently as a separate timer without being tuned. In this case, the match interrupt signal of TAAn cannot be used as a trigger source to start A/D conversion in the 6-phase PWM output mode. Therefore, fix the TABnOPT2.TABnAT2, TABnOPT2.TABnAT3, TABnOPT3.TABnAT6, and TABnOPT3.TABnAT7 bits to 0.

The other control bits can be used in the same manner as when TAAn is tuned.

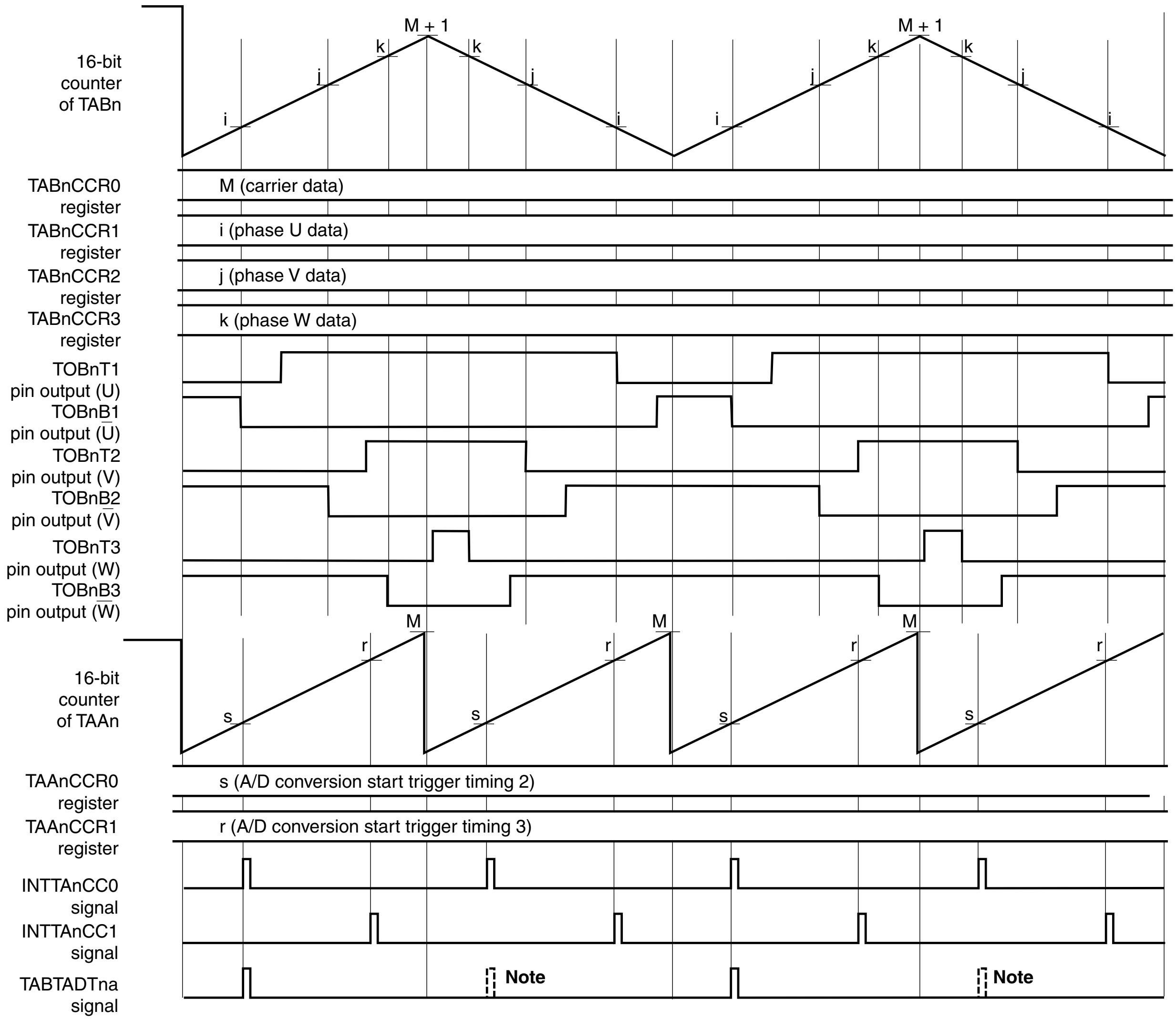
If TAAn is not tuned, the compare registers (TAAnCCR0 and TAAnCCR1) of TAAn are not affected by the settings of the TABnOPT0.TABnCMS and TABnOPT2.TABnRDE bits. For the initialization procedure when TAAn is not tuned, see (b) to (e) in **10.4.5 (1) Tuning operation starting procedure**. (a) is not necessary because it is a step used to set TAAn for the tuning operation.

(4) Basic operation of TAAn during tuning operation

The 16-bit counter of TAAn only counts up. The 16-bit counter is cleared by the set cycle value of the TABnCCR0 register and starts counting from 0000H again. The count value of this counter is the same as the value of the 16-bit counter of TAAn when it counts up. However, it is not the same when the 16-bit counter of TABn counts down.

- When TABn counts up (same value)
  16-bit counter of TABn: 0000H → M (up counting)
  16-bit counter of TAAn: 0000H → M (up counting)
- When TABn counts down (not same value)
  16-bit counter of TABn: M + 1 → 0001H (down counting)
  16-bit counter of TAAn: 0000H → M (up counting)

Figure 10-37. TAAn During Tuning Operation



Note The TABTADTn0 signal is masked by the TABnOPT2.TABnATM2 and TABnOPT2.TABnATM3 bits. The TABTADTn1 signal is masked by the TABnOPT3.TABnATM6 and TABnOPT3.TABnATM7 bits.

Remark V850E/IG4-H: n = 0, a = 0, 1
V850E/IH4-H: n = 0, 1, a = 0, 1

10.4.6 A/D conversion start trigger output function

The V850E/IG4-H and V850E/IH4-H have a function to select four trigger sources (INTTBnOV, INTTBnCC0, INTTAnCC0, INTTAnCC1) to generate the A/D conversion start trigger signal (TABTADTn0, TABTADTn1) of A/D converters 0 and 1.

The trigger sources are specified by the TABnOPT2.TABnAT0 to TABnOPT2.TABnAT3 and TABnOPT3.TABnAT4 to TABnOPT3.TABnAT7 bits.

- TABnAT0, TABnAT4 bits = 1:
  A/D conversion start trigger signal generated when INTTBnOV (counter underflow) occurs.
- TABnAT1, TABnAT5 bits = 1:
  A/D conversion start trigger signal generated when INTTBnCC0 (cycle match) occurs.
- TABnAT2, TABnAT6 bits = 1:
  A/D conversion start trigger signal generated when INTTAnCC0 (match of TAAnCCR0 register of TAAn during tuning operation) occurs.
- TABnAT3, TABnAT7 bits = 1:
  A/D conversion start trigger signal generated when INTTAnCC1 (match of TAAnCCR1 register of TAAn during tuning operation) occurs.

The A/D conversion start trigger signals selected by the TABnAT0 to TABnAT3 and TABnAT4 to TABnAT7 bits are ORed and output. Therefore, two or more trigger sources can be specified at the same time.

The INTTBnOV and INTTBnCC0 signals selected by the TABnAT0, TABnAT1, TABnAT4, and TABnAT5 bits are culled interrupt signals.

Therefore, these signals are output after the interrupts have been culled and, unless interrupt output is enabled (TABnOPT1.TABnICE, TABnOPT1.TABnIOE bits), the A/D conversion start trigger is not output.

The trigger sources (INTTAnCC0 and INTTAnCC1) from TAAn have a function to mask the A/D conversion start trigger signal depending on the status of the up-count/down-count of the 16-bit counter, if so set by the TABnAT2, TABnAT3, TABnAT6, and TABnAT7 bits.

- TABnATM2, TABnATM6 bits:
  Correspond to the TABnAT2 and TABnAT6 bits and control INTTAnCC0 (match interrupt signal) of TAAn.
  - TABnATM2, TABnATM6 bits = 0
    The A/D conversion start trigger signal is output when the 16-bit counter counts up (TABnOPT0.TABnCUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TABnOPT0.TABnCUF bit = 1).
  - TABnATM2, TABnATM6 bits = 1
    The A/D conversion start trigger signal is output when the 16-bit counter counts down (TABnOPT0.TABnCUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts up (TABnOPT0.TABnCUF bit = 0).

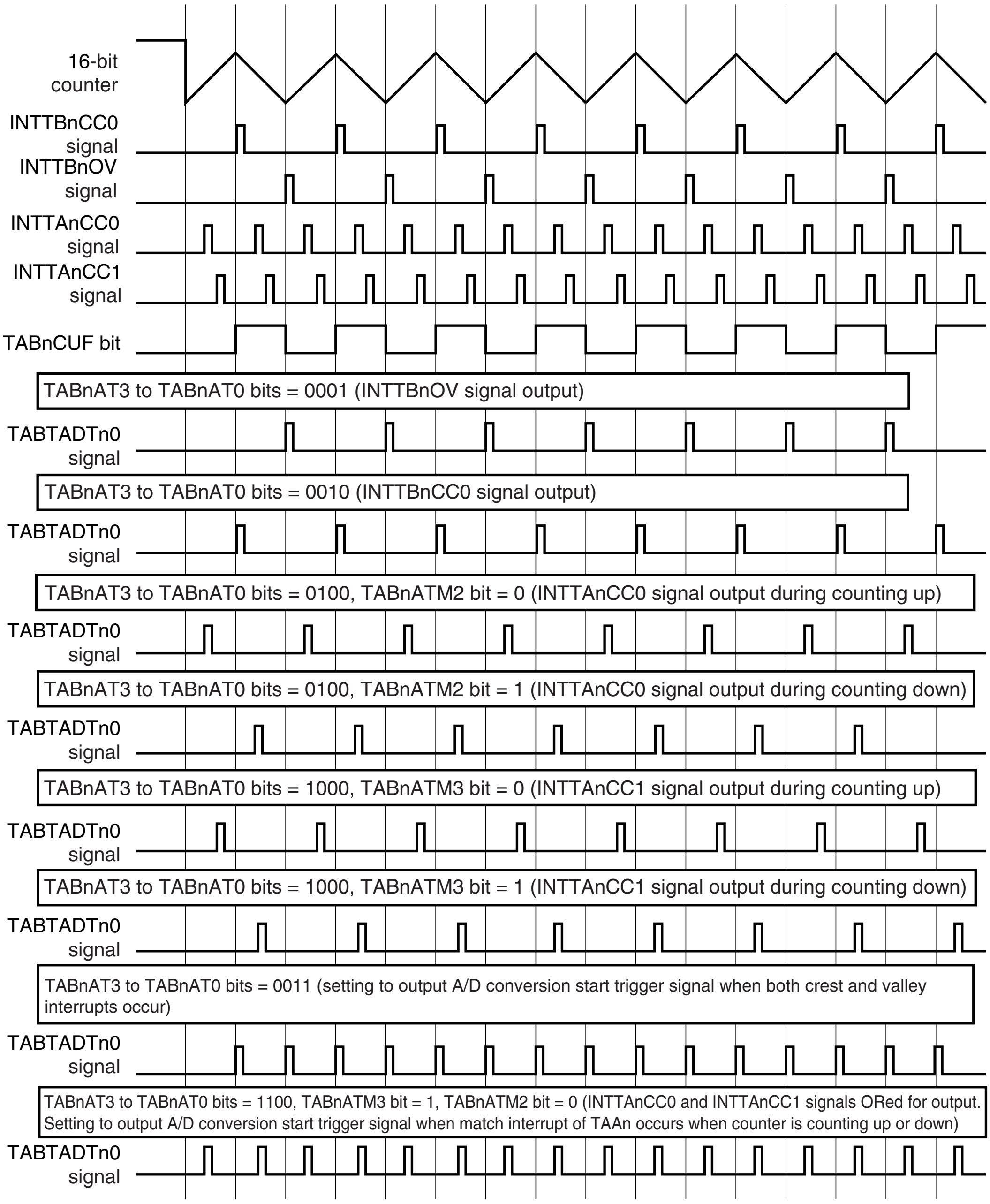
- TABnATM3, TABnATM7 bits:
  Correspond to the TABnAT3 and TABnAT7 bits and control INTTAnCC1 (match interrupt signal) of TAAn.
  - TABnATM3, TABnATM7 bits = 0
    The A/D conversion start trigger signal is output when the 16-bit counter counts up (TABnCUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TABnCUF bit = 1).
  - TABnATM3, TABnATM7 bits = 1
    The A/D conversion start trigger signal is output when the 16-bit counter counts down (TABnCUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts up (TABnCUF bit = 0).

The TABnATM3, TABnATM2, TABnAT3 to TABnAT0, TABnATM7, TABnATM6, and TABnAT7 to TABnAT4 bits can be rewritten while the timer is operating. If the bit that sets the A/D conversion start trigger signal is rewritten while the timer is operating, the new setting is immediately reflected on the output status of the A/D conversion start trigger. These control bits do not have a transfer function and can be used only in the anytime rewriting mode.

Cautions 1. The A/D conversion start trigger signal output that is set by the TABnAT2, TABnAT3, TABnAT6, and TABnAT7 bits can be used only when TAAn is performing a tuning operation as the slave timer of TABn. If TABn and TAAn are not performing a tuning operation, or if a mode other than the 6-phase PWM output mode is used, the output cannot be guaranteed.

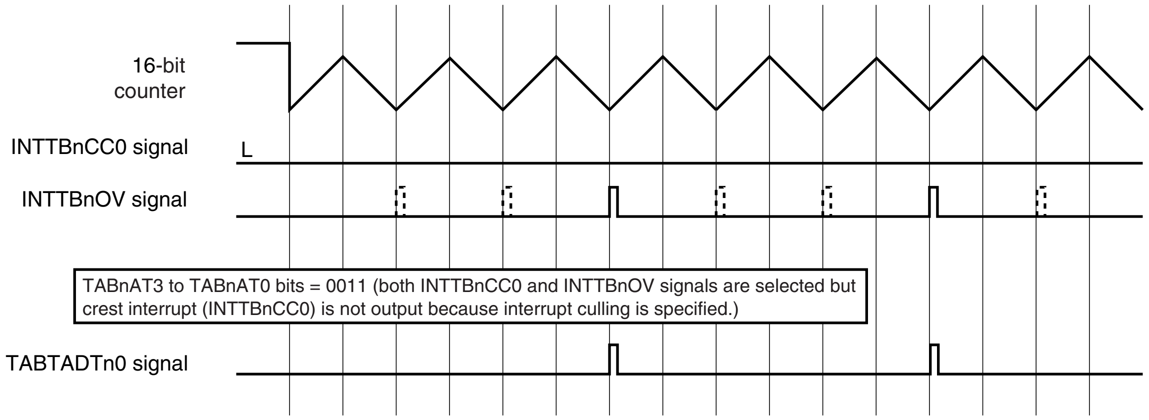
2. The TOBn0 signal output is internally used to identify whether the 16-bit counter is counting up or down. Therefore, enable TOBn0 pin output by setting the TABnIOC0.TABnOL0 bit to 0 and the TABnIOC0.TABnOE0 bit to 1.

Figure 10-38. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00000: Without Interrupt Culling)


16-bit counter
INTTBnCC0 signal
INTTBnOV signal
INTTAnCC0 signal
INTTAnCC1 signal
TABnCUF bit
TABnAT3 to TABnAT0 bits = 0001 (INTTBnOV signal output)
TABTADTn0 signal
TABnAT3 to TABnAT0 bits = 0010 (INTTBnCC0 signal output)
TABTADTn0 signal
TABnAT3 to TABnAT0 bits = 0100, TABnATM2 bit = 0 (INTTAnCC0 signal output during counting up)
TABTADTn0 signal
TABnAT3 to TABnAT0 bits = 0100, TABnATM2 bit = 1 (INTTAnCC0 signal output during counting down)
TABTADTn0 signal
TABnAT3 to TABnAT0 bits = 1000, TABnATM3 bit = 0 (INTTAnCC1 signal output during counting up)
TABTADTn0 signal
TABnAT3 to TABnAT0 bits = 1000, TABnATM3 bit = 1 (INTTAnCC1 signal output during counting down)
TABTADTn0 signal
TABnAT3 to TABnAT0 bits = 0011 (setting to output A/D conversion start trigger signal when both crest and valley interrupts occur)
TABTADTn0 signal
TABnAT3 to TABnAT0 bits = 1100, TABnATM3 bit = 1, TABnATM2 bit = 0 (INTTAnCC0 and INTTAnCC1 signals ORed for output. Setting to output A/D conversion start trigger signal when match interrupt of TAAn occurs when counter is counting up or down)
TABTADTn0 signal

Remark V850E/IG4-H: n = 0
V850E/IH4-H: n = 0, 1

Figure 10-39. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00010: With Interrupt Culling) (1)



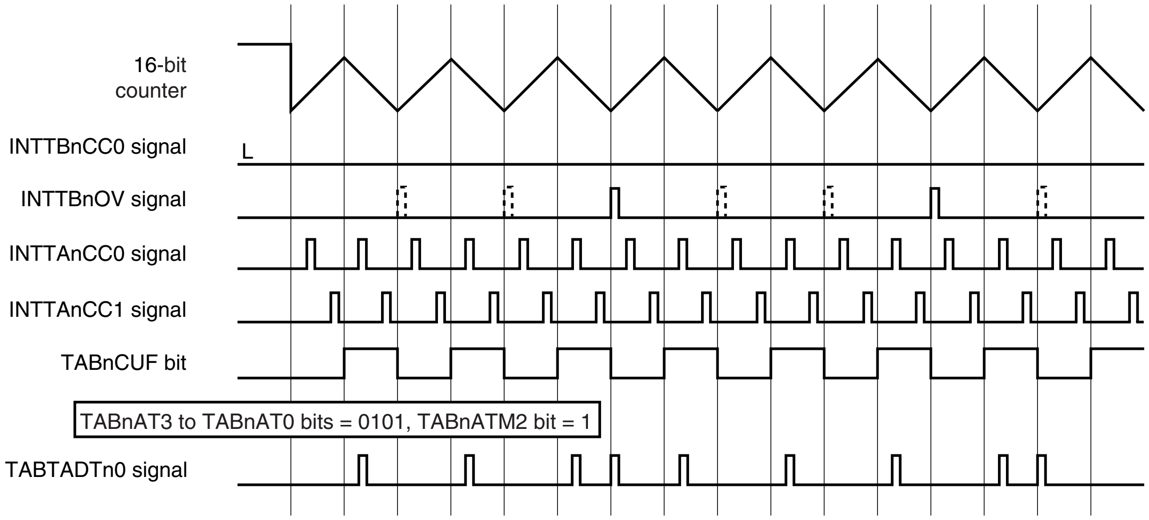
TABnAT3 to TABnAT0 bits = 0011 (both INTTBnCC0 and INTTBnOV signals are selected but crest interrupt (INTTBnCC0) is not output because interrupt culling is specified.)

Remarks 1. ⊓: Culled interrupt

2. V850E/IG4-H: n = 0

V850E/IH4-H: n = 0, 1

Figure 10-40. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00010: With Interrupt Culling) (2)



TABnAT3 to TABnAT0 bits = 0101, TABnATM2 bit = 1

Caution The INTTBnCC0 signal is culled but the INTTAnCC0 signal is not.

Remarks 1. ⊓: Culled interrupt

2. V850E/IG4-H: n = 0

V850E/IH4-H: n = 0, 1

(1) Operation under boundary condition (operation when 16-bit counter matches INTTAnCC0 signal)

Table 10-3. Operation When TABnCCR0 Register = M, TABnAT2 Bit = 1, TABnAT6 Bit = 1, TABnATM2 Bit = 0, TABnATM6 Bit = 0 (Up Counting Period Selected)

Value of TAAnCCR0 Register	Value of 16-bit Counter of TABn	Value of 16-bit Counter of TAAn	Status of 16-bit Counter of TABn	Output of INTTAnCC0 Signal from TABTADTna Signal
0000H	0000H	0000H	–	Output
0000H	M + 1	0000H	–	Not output
0001H	0001H	0001H	Up count	Output
0001H	M	0001H	Down count	Not output
M	M	M	Up count	Output
M	0001H	M	Down count	Not output

Table 10-4. Operation When TABnCCR0 Register = M, TABnAT2 Bit = 1, TABnAT6 Bit = 1, TABnATM2 Bit = 1, TABnATM6 Bit = 1 (Down Counting Period Selected)

Value of TAAnCCR0 Register	Value of 16-bit Counter of TABn	Value of 16-bit Counter of TAAn	Status of 16-bit Counter of TABn	Output of INTTAnCC0 Signal from TABTADTna Signal
0000H	0000H	0000H	–	Not output
0000H	M + 1	0000H	–	Output
0001H	0001H	0001H	Up count	Not output
0001H	M	0001H	Down count	Output
M	M	M	Up count	Not output
M	0001H	M	Down count	Output

Caution The TAAnCCRa register enables setting of “0” to “M” when the TABnCCR0 register = M. Setting of a value of “M + 1” or higher is prohibited.
If a value higher than “M + 1” is set, the 16-bit counter of TAAn is cleared by “M”. Therefore, the TABTADTna signal is not output.

Remark V850E/IG4-H: n = 0, a = 0, 1
V850E/IH4-H: n = 0, 1, a = 0, 1

CHAPTER 11 WATCHDOG TIMER FUNCTIONS

11.1 Functions

The watchdog timer has the following functions.

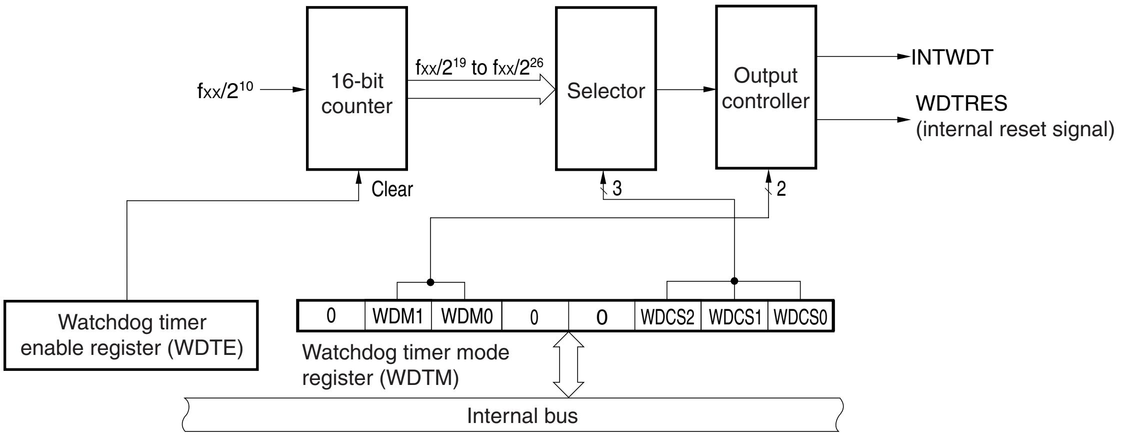
- Reset mode: Reset operation upon overflow of the watchdog timer (generation of WDTRES signal)
- Non-maskable interrupt request mode:
  Non-maskable interrupt operation upon overflow of the watchdog timer (generation of INTWDT signal)

Caution The watchdog timer is stopped after reset is released.
It starts operating when “ACH” is written to the WDTE register. Also, write to the WDTM register for verification purposes only once, even if the default settings (reset mode, interval time: 2^{26}/fxx) do not need to be changed.

11.2 Configuration

The block diagram of the watchdog timer is shown below.

Figure 11-1. Block Diagram of Watchdog Timer



Remark fxx/2^{10}: Watchdog timer clock
fxx: Peripheral clock
INTWDT: Non-maskable interrupt request signal upon overflow of watchdog timer
WDTRES: Reset signal upon overflow of watchdog timer

The watchdog timer consists of the following hardware.

Table 11-1. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer mode register (WDTM) Watchdog timer enable register (WDTE)

11.3 Control Registers

(1) Watchdog timer mode register (WDTM)

The WDTM register sets the overflow time and operation clock of the watchdog timer.

This register can be read or written in 8-bit units. This register can be read any number of times, but can be written only once following reset release; it cannot then be written a second or subsequent time.

Reset sets this register to 67H.

After reset: 67H R/W Address: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM	0	WDM1	WDM0	0	0	WDCS2	WDCS1	WDCS0

WDM1	WDM0	Selection of operation mode of watchdog timer
0	0	Stop operation
0	1	Non-maskable interrupt request mode (generation of INTWDT signal)
1	×	Reset mode (generation of WDTRES signal)

Cautions 1. For details of the WDCS2 to WDCS0 bits, see Table 11-2 Overflow Time.

2. If the WDTM register is rewritten while the watchdog timer is counting, the counter of the watchdog timer is cleared to 0000H.

3. Be sure to clear bits 3, 4, and 7 to “0”.

Table 11-2. Overflow Time

WDCS2	WDCS1	WDCS0	Overflow Time	f_{xx} = 100 MHz
0	0	0	$2^{19}/f_{xx}$	5.2 ms
0	0	1	$2^{20}/f_{xx}$	10.5 ms
0	1	0	$2^{21}/f_{xx}$	21.0 ms
0	1	1	$2^{22}/f_{xx}$	41.9 ms
1	0	0	$2^{23}/f_{xx}$	83.9 ms
1	0	1	$2^{24}/f_{xx}$	167.8 ms
1	1	0	$2^{25}/f_{xx}$	335.5 ms
1	1	1	$2^{26}/f_{xx}$	671.1 ms

(2) Watchdog timer enable register (WDTE)

The counter of the watchdog timer is cleared and counting restarted by writing “ACH” to the WDTE register.

This register can be read or written in 8-bit units.

Reset sets this register to 1AH.

After reset: 1AH R/W Address: FFFFF6D1H

	7	6	5	4	3	2	1	0
WDTE								

Cautions 1. If “ACH” is written to the WDTE register to enable the watchdog timer operation and then a value other than “ACH” is written to the WDTE register, a non-maskable interrupt request signal (INTWDT) or a reset signal (WDTRES) is generated due to watchdog timer overflow, depending on the specification of the WDTM.WDM1 and WDTM.WDM0 bits.

2. When the WDTE register is read or written in 1-bit units, an internal reset signal is output.

3. The read value of the WDTE register is “1AH” before the watchdog timer operates, and “9AH” after it operates. The value read from this register is different from the written value (ACH).

11.4 Operation

The watchdog timer is stopped after reset is released.

The WDTM register can be written only once after reset ends.

To use the watchdog timer, write the operation mode and the interval time to the WDTM register in 8-bit units. After this, the operation of the watchdog timer cannot be stopped.

To not use the watchdog timer, write 00H to the WDTM register.

11.5 Caution

The cycle of the non-maskable interrupt request signal (INTWDT) that is generated due to watchdog timer overflow can be calculated from “Interval time set to WDTM register + 2^7 peripheral clock pulse width”, if INTWDT occurs successively without the watchdog timer being cleared.

Note that the pulse width until generation of the first interrupt request signal after the watchdog timer has been started is not included.

CHAPTER 12 A/D CONVERTERS 0 AND 1

12.1 Features

○ Two 12-bit resolution A/D converter circuits (A/D converters 0 and 1)
Simultaneous sampling of two circuits possible

○ Analog input

- When the comparator is not used
  [V850E/IG4-H]
  Total of 7 channels in two circuits
  A/D converter 0: ANI00/ANI05, ANI01/ANI06, ANI02/ANI07, ANI03 (4 channels)
  A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17 (3 channels)
  [V850E/IH4-H]
  Total of 8 channels in two circuits
  A/D converter 0: ANI00/ANI05, ANI01/ANI06, ANI02/ANI07, ANI03 (4 channels)
  A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17, ANI13 (4 channels)
- When the comparator is used
  Total of 6 channels in two circuits
  [when the low-range and full-range comparators are used]
  A/D converter 0: ANI00/ANI05, ANI01/ANI06, ANI02/ANI07 (3 channels)
  A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17 (3 channels)

○ A/D conversion result registers
12 bits × 16 + 12 bits × 16
A/D converter 0: AD0CR0 to AD0CR15
A/D converter 1: AD1CR0 to AD1CR15

○ A/D conversion result extension registers
Can be used only in the extension buffer mode
12 bits × 5 + 12 bits × 5
A/D converter 0: AD0ECR0 to AD0ECR4
A/D converter 1: AD1ECR0 to AD1ECR4

○ Operation modes

- Normal operation modes
  A/D trigger mode
  A/D trigger polling mode
  Hardware trigger mode
- Extension operation modes
  Conversion channel specification mode
  Extension buffer mode

○ Operational amplifiers for input level amplification (×2.5 to ×10)
These channels can be used only when the operational amplifier for input level amplification is used.
Total of 6 units in two circuits
A/D converter 0: ANI05 to ANI07 (3 units)
A/D converter 1: ANI15 to ANI17 (3 units)

○ Overvoltage detection comparator

- These channels can be used only when the overvoltage detection comparator is used.
- Total of 6 units in two circuits
  A/D converter 0: 3 units
  A/D converter 1: 3 units
- Reference voltage
  The reference voltage is generated by using on-chip 8-bit D/A converters 0 and 1.
- An interrupt occurs when an overvoltage is detected. Interrupt requests are output by using the two output signals (for full range and low range) generated after ORing or ANDing overvoltage detection signals input from the ANI00/ANI05, ANI01/ANI06, and ANI02/ANI07 channels (A/D converter 0) or the two output signals that are generated after ORing or ANDing overvoltage detection signals input from the ANI10/ANI15, ANI11/ANI16, and ANI12/ANI17 channels (A/D converter 1).
- The output of a timer for motor control can be set to a high-impedance state when an overvoltage is detected.

○ Successive approximation method

○ Operating voltage range

$EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3}$ (V850E/IH4-H only) $= AV_{DD0} = AV_{DD1} = AV_{REFP0} = AV_{REFP1} = 4.0$ to 5.5 V

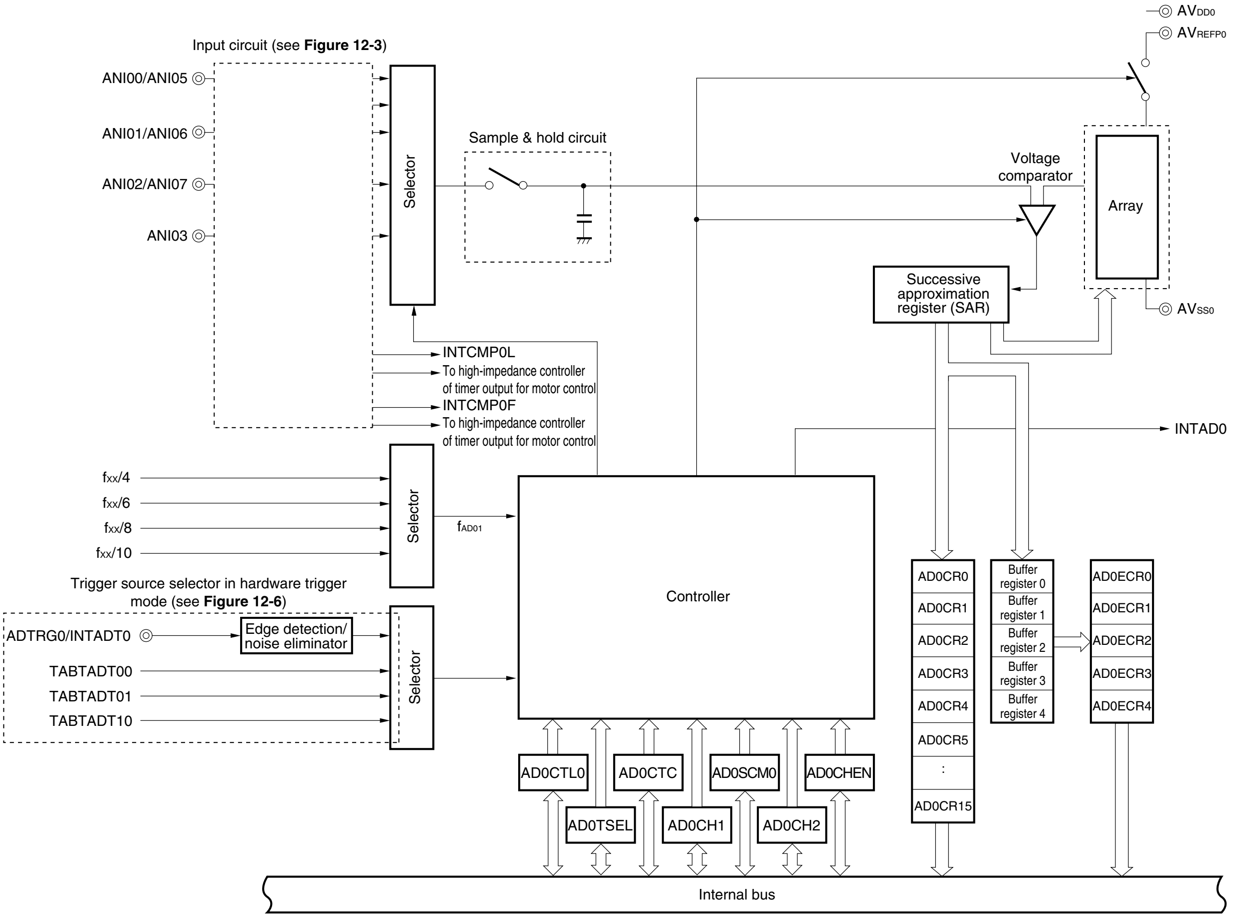
○ 8-bit D/A converters 0 and 1

- Total of 4 channels in two circuits
  D/A converter 0: 2 channels
  D/A converter 1: 2 channels
- No external pins or alternate-function port pins
- They operate only in the normal operating mode. (The real-time output mode is not available.)
- The reference voltage supplied to the comparators in the A/D converters is generated by:
  Low-range reference voltage: D/A converter 00, D/A converter 10
  Full-range reference voltage: D/A converter 01, D/A converter 11
- Settling time: 10 μs

12.2 Configuration

The block diagram is shown below.

Figure 12-1. Block Diagram of A/D Converter 0

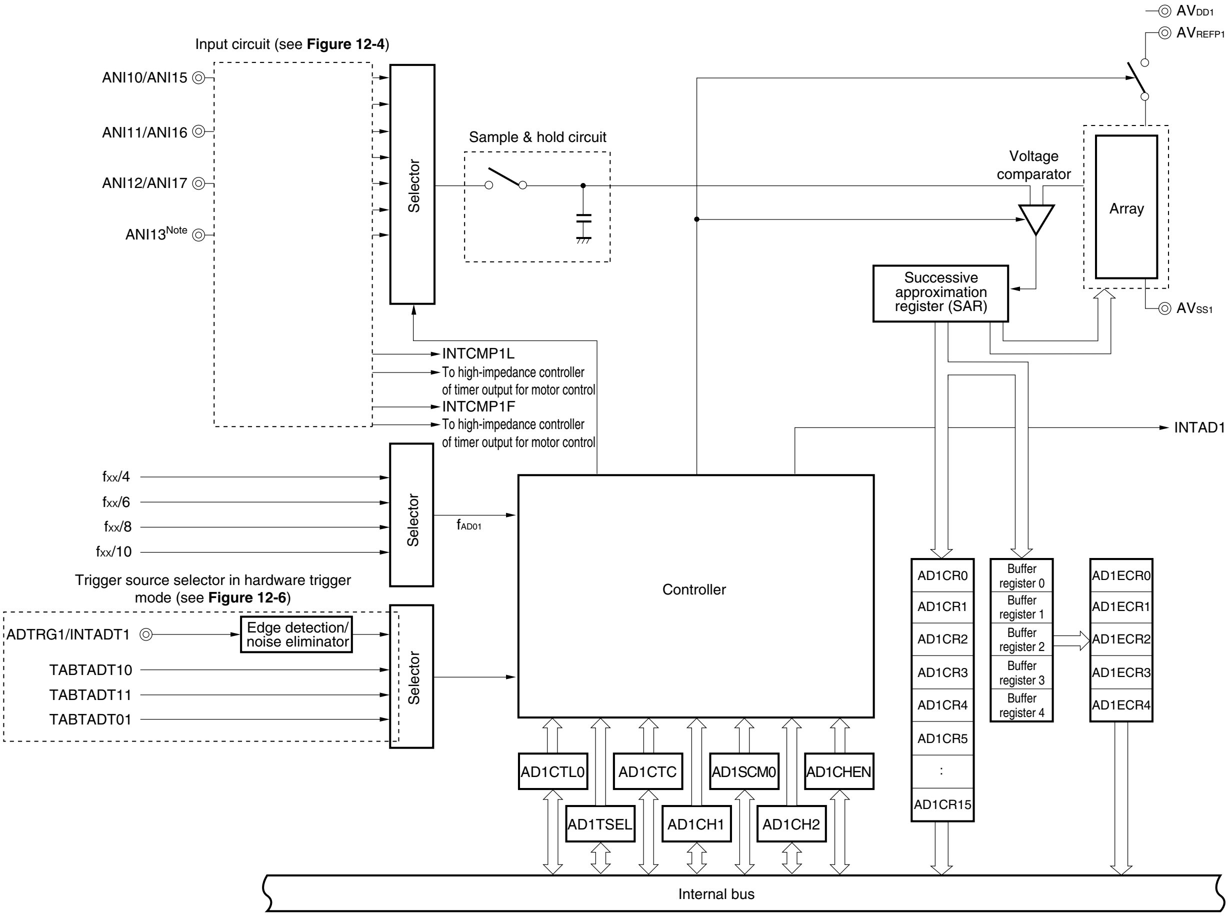


Remark fXX: Peripheral clock

fAD01: Base clock

Buffer registers 0 to 4: A/D0 conversion result extension buffer registers 0 to 4

Figure 12-2. Block Diagram of A/D Converter 1



Note V850E/IH4-H only

Remark fxx: Peripheral clock

fAD01: Base clock

Buffer registers 0 to 4: A/D1 conversion result extension buffer registers 0 to 4

Cautions 1. If there is noise at the analog input pins (ANI00 to ANI03, ANI05 to ANI07, ANI10 to ANI12, ANI13 (V850E/IH4-H only), ANI15 to ANI17) or at the A/D converter reference voltage input pins (AV_{REFP0}, AV_{REFP1}), that noise may generate an illegal conversion result.

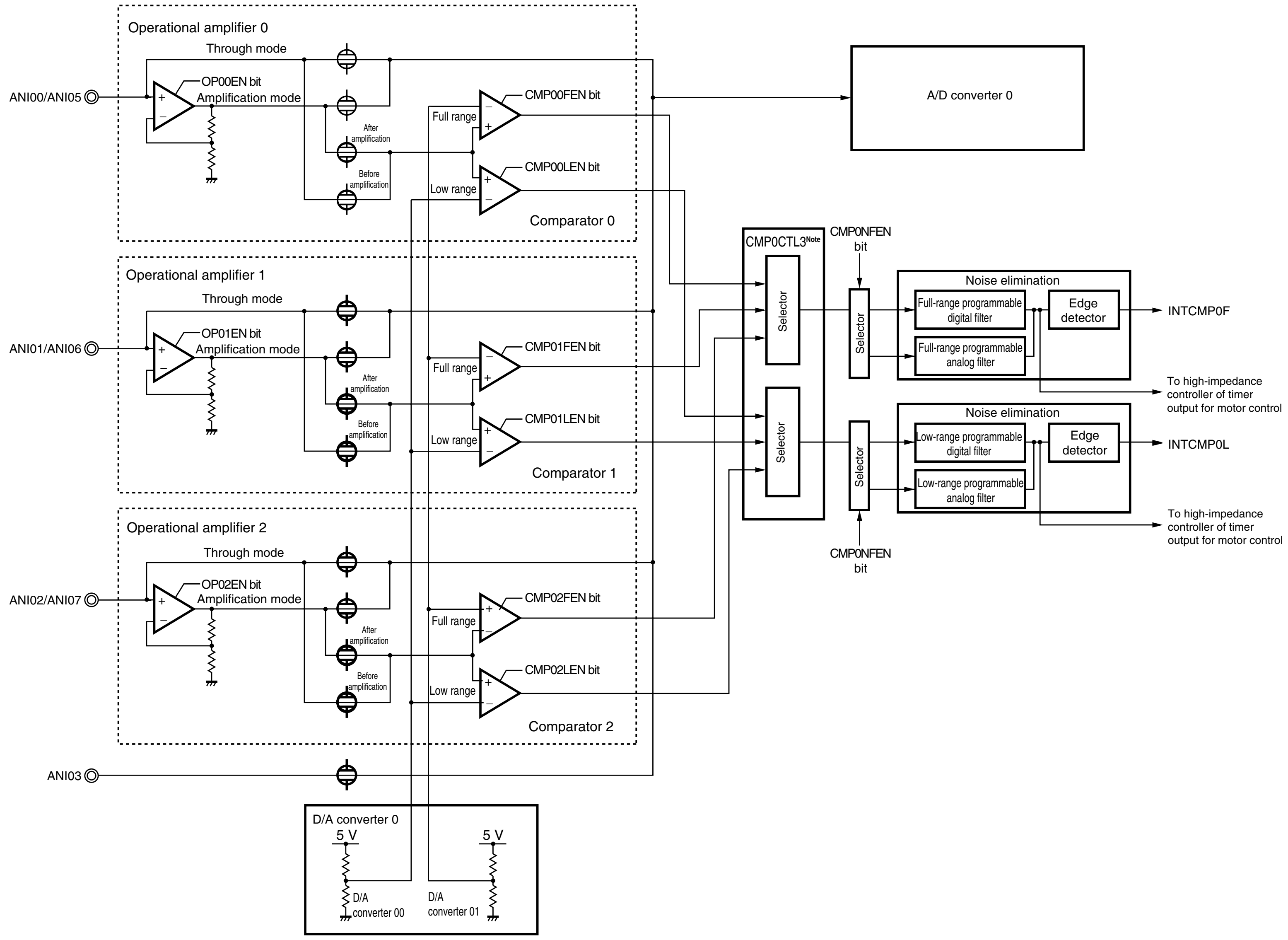
Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.

An example of this software processing is shown below.

- **Take the average result of a number of A/D conversions and use that as the A/D conversion result.**
- **Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.**
- **If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.**

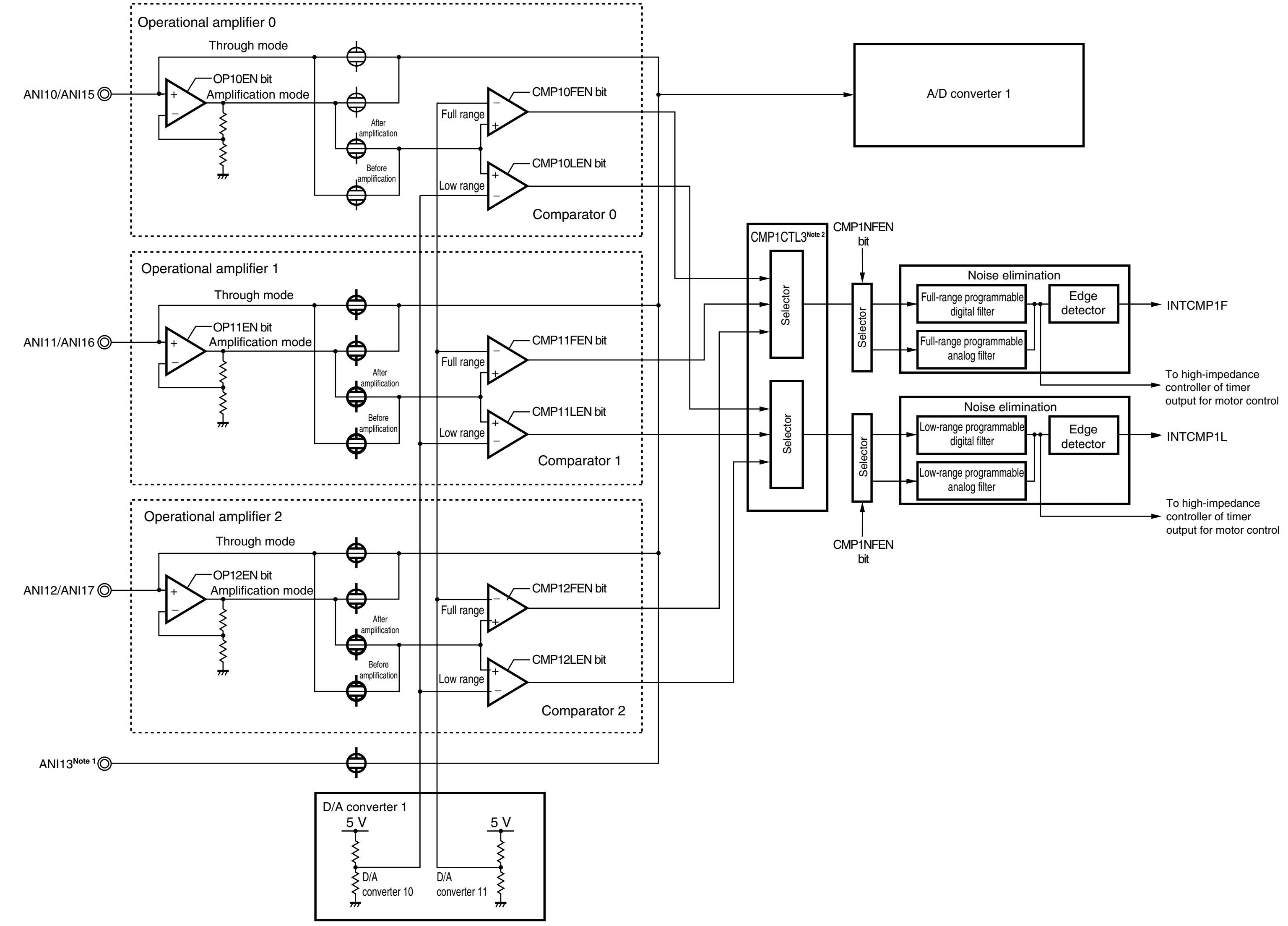
2. Do not apply a voltage outside the AV_{SSn} to AV_{REFPn} range to the pins that are used as input pins of A/D converters 0 and 1 (n = 0, 1).

Figure 12-3. Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comparator in A/D Converter 0



Note For details, see **Figure 12-5 CMPnCTL3 Register Selector Circuit Configuration**.

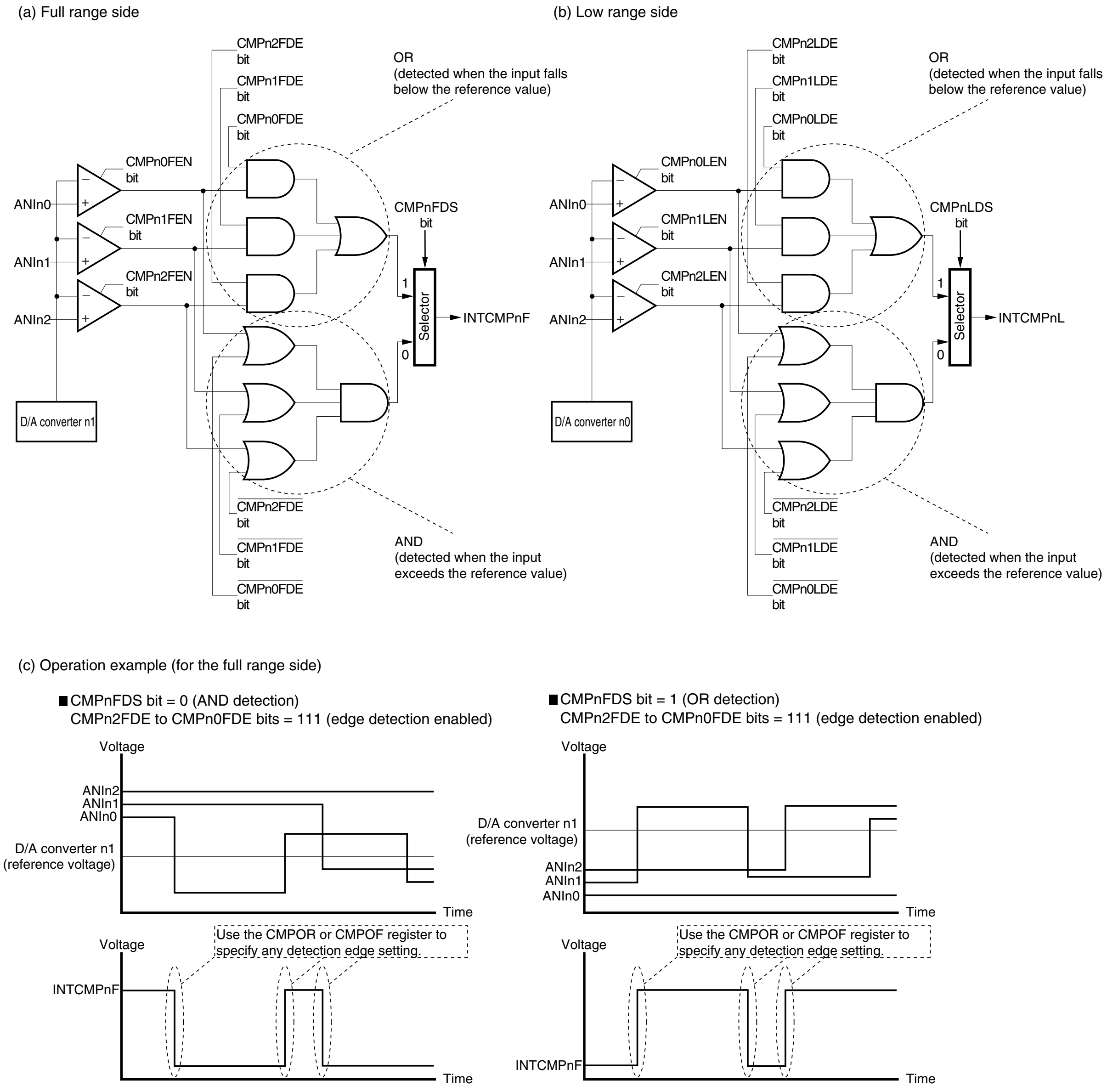
Figure 12-4. Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comparator in A/D Converter 1



Notes 1. V850E/IH4-H only

2. For details, see **Figure 12-5 CMPnCTL3 Register Selector Circuit Configuration**.

Figure 12-5. CMPnCTL3 Register Selector Circuit Configuration



(a) Full range side

(b) Low range side

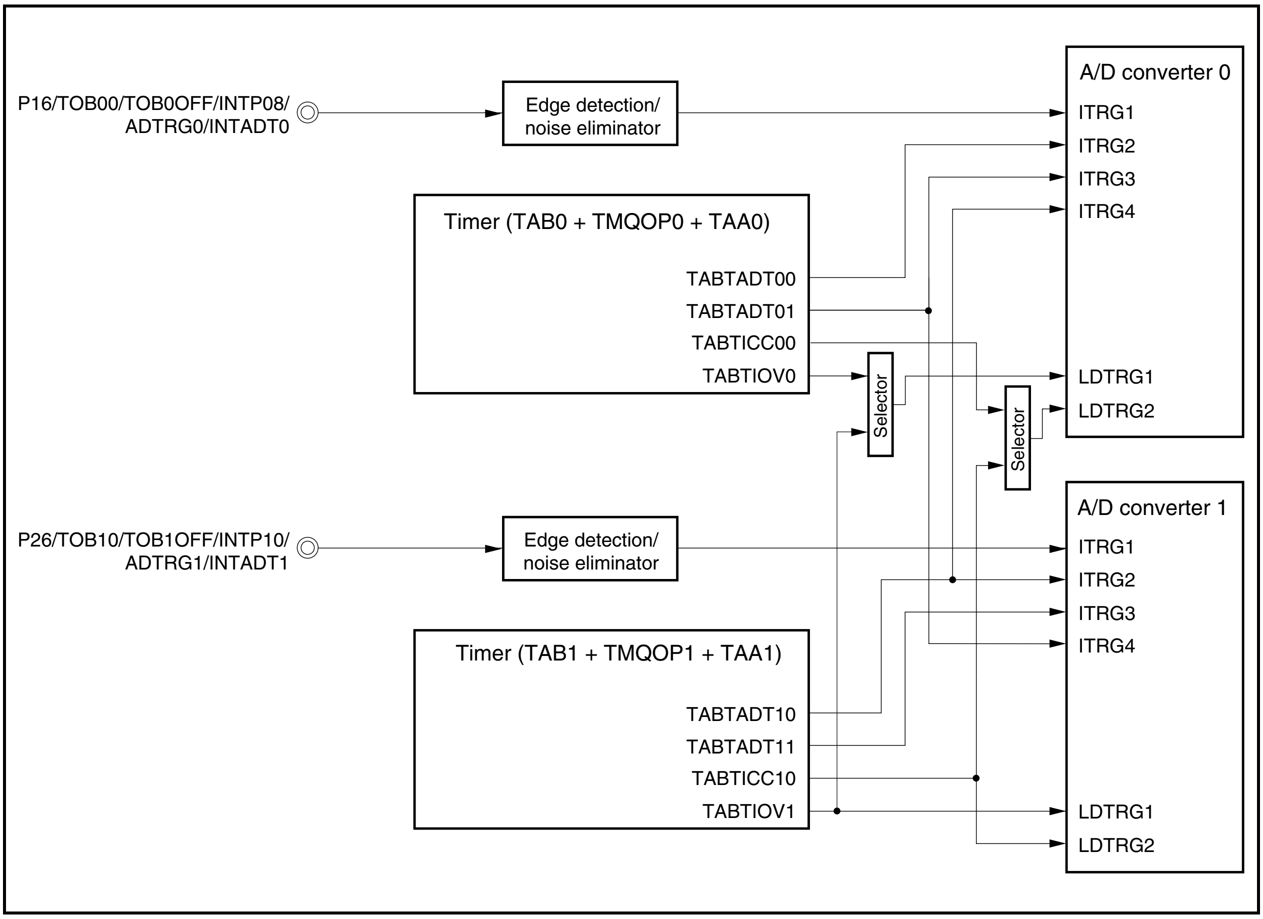
(c) Operation example (for the full range side)

CMPn2FDE bit
CMPn1FDE bit
CMPn0FDE bit
OR
(detected when the input falls below the reference value)
CMPn0FEN bit
CMPn1FEN bit
CMPn2FEN bit
ANIn0
ANIn1
ANIn2
CMPnFDS bit
1
0
Selector
INTCMPnF
D/A converter n1
CMPn2FDE bit
CMPn1FDE bit
CMPn0FDE bit
AND
(detected when the input exceeds the reference value)
CMPn2LDE bit
CMPn1LDE bit
CMPn0LDE bit
OR
(detected when the input falls below the reference value)
CMPn0LEN bit
CMPn1LEN bit
CMPn2LEN bit
CMPnLDS bit
INTCMPnL
D/A converter n0
CMPn2LDE bit
CMPn1LDE bit
CMPn0LDE bit
AND
(detected when the input exceeds the reference value)
■CMPnFDS bit = 0 (AND detection)
CMPn2FDE to CMPn0FDE bits = 111 (edge detection enabled)
■CMPnFDS bit = 1 (OR detection)
CMPn2FDE to CMPn0FDE bits = 111 (edge detection enabled)
Voltage
ANIn2
ANIn1
ANIn0
D/A converter n1
(reference voltage)
Time
Use the CMPOR or CMPOF register to specify any detection edge setting.
INTCMPnF

Remarks 1. n = 0, 1

2. Details about the noise eliminator have been omitted from the description.

Figure 12-6. Block Diagram of Trigger Source Selector in Hardware Trigger Mode



A/D converters 0 and 1 consist of the following hardware.

Table 12-1. Configuration of A/D Converters 0 and 1 (1/2)

Item	Configuration
Analog input	[V850E/IG4-H] When comparator is not used: ANI00/ANI05, ANI01/ANI06, ANI02/ANI07, ANI03, ANI10/ANI15, ANI11/ANI16, ANI12/ANI17 (Total of 7 channels in two circuits) When comparator is used (when the low-range and full-range comparators are used): ANI00/ANI05, ANI01/ANI06, ANI02/ANI07, ANI10/ANI15, ANI11/ANI16, ANI12/ANI17 (Total of 6 channels in two circuits) [V850E/IH4-H] When comparator is not used: ANI00/ANI05, ANI01/ANI06, ANI02/ANI07, ANI03, ANI10/ANI15, ANI11/ANI16, ANI12/ANI17, ANI13 (Total of 8 channels in two circuits) When comparator is used (when the low-range and full-range comparators are used): ANI00/ANI05, ANI01/ANI06, ANI02/ANI07, ANI10/ANI15, ANI11/ANI16, ANI12/ANI17 (Total of 6 channels in two circuits)
Registers	Successive approximation register (SAR) A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15) A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H) A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4) (only in extension operation mode (extension buffer mode)) A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H) (only in extension operation mode (extension buffer mode))
Control registers	A/D converter n scan mode register (ADnSCM) A/D converter n scan mode register L (ADnSCML) A/D converter n scan mode register H (ADnSCMH) A/D converter n conversion time control register (ADnCTC) A/D converter n conversion channel specification register (ADnCHEN) A/D converter n conversion channel specification register L (ADnCHENL) A/D converter n conversion channel specification register H (ADnCHENH) A/D converter n control register (ADnCTL0) A/D converter n trigger select register (ADnTSEL) A/D converter n channel specification register 1 (ADnCH1) A/D converter n channel specification register 2 (ADnCH2) A/D converter n flag register (ADnFLG) A/D converter n flag buffer register (ADnFLGB) A/DLDTRG1 input select register (ADLTS1) A/DLDTRG2 input select register (ADLTS2) A/D converter n clock select register (ADnOCKS) A/D trigger falling edge specification register (ADTF) A/D trigger rising edge specification register (ADTR) Operational amplifier n control register 0 (OPnCTL0) Comparator n control register 0 (CMPnCTL0) Comparator n control register 1 (CMPnCTL1) Comparator n control register 2 (CMPnCTL2) Comparator n control register 3 (CMPnCTL3) Comparator output digital noise elimination register nL (CMPNFCnL) Comparator output digital noise elimination register nF (CMPNFCnF) Comparator output interrupt rising edge specification register (CMPOR) Comparator output interrupt falling edge specification register (CMPOF) D/A converter n mode register (DAnM) D/A converter n conversion value setting registers 0, 1 (DAnCS0, DAnCS1)

Remark n = 0, 1

(1) Selector

The selector selects the analog input pin according to the mode set by the ADnSCM, ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, ADLTS2, and ADnOCKS registers and sends the input to the sample & hold circuit (n = 0, 1).

ANI05 to ANI07, ANI15 to ANI17 are provided with an operational amplifier for input level amplification and an overvoltage detection comparator. The operational amplifier and comparator of each analog input pin can be specified to be on or off. The amplification (gain) of the operational amplifier can be selected from 2.5 to 10 times for ANI05 to ANI07, ANI15 to ANI17.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the voltage comparator. When the operational amplifier for input level amplification is used, the gain specified by the OPnCTL0.OPnGA3 to OPnCTL0.OPnGA0 bits × the input voltage is sampled. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) Voltage comparator

This comparator compares the voltage generated from the voltage tap of the array with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REFPn}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REFPn}), the MSB of the SAR is reset.

After that, bit 10 of the SAR is automatically set, and the next comparison is made. The voltage tap of the array is selected by the value of bit 11, to which the result has been already set.

Bit 11 = 0: (1/4 AV_{REFPn})
Bit 11 = 1: (3/4 AV_{REFPn})

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 10 = 1
Analog input voltage ≤ Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR.

(4) Array

The array generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR is a 12-bit register that sets voltage tap data whose values from the array match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR (conversion results) are held in A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15) (n = 0, 1). In the extension buffer mode, however, the conversion result is stored in A/Dn conversion result extension buffer registers 0 to 4 and, when selection load trigger x is generated, shifted to and stored in the ADnECR0 to ADnECR4 registers (x = 1, 2). When all the specified A/D conversion operations have ended, an A/Dn conversion end interrupt request signal (INTADn) is generated.

(6) A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15), A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H) (n = 0, 1)

The ADnCR0 to ADnCR15 and ADnCR0H to ADnCR15H registers are registers that hold the A/D conversion results. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 12 bits of the ADnCR0 to ADnCR15 registers. The lower 4 bits of these registers are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the ADnCR0H to ADnCR15H registers.

To read the result of A/D conversion in 16-bit units, specify the ADnCR0 to ADnCR15 registers. To read the higher 8 bits, specify the ADnCR0H to ADnCR15H registers.

(7) A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4), A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H) (n = 0, 1)

The ADnECR0 to ADnECR4 and ADnECR0H to ADnECR4H registers are registers that hold the A/D conversion results. These registers can be used only in extension buffer mode. When A/D conversion is completed, the A/D conversion result is stored in the A/Dn conversion result extension buffer register. If selection load trigger 1 is generated after that, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 0 to 2 to the higher 12 bits of the ADnECR0 to ADnECR2 registers for storage. Bits 1 to 3 are always 0 when read. If selection load trigger 2 is generated, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 3 and 4 to the higher 12 bits of the ADnECR3 and ADnECR4 registers. Bits 1 to 3 are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the ADnECR0H to ADnECR4H registers.

To read the result of A/D conversion in 16-bit units, specify the ADnECR0 to ADnECR4 registers. To read the higher 8 bits, specify the ADnECR0H to ADnECR4H registers.

(8) ANIn0 to ANIn3, ANIn5 to ANIn7 pins (n = 0, 1)

The ANIn0 to ANIn3 and ANIn5 to ANIn7 pins (ANI10 to ANI12 and ANI15 to ANI17 pins only in A/D converter 1 of V850E/IG4-H) are analog input pins for A/D converters 0 and 1. They input the analog signals to be A/D converted.

Caution Make sure that the voltages input to the ANIn0 to ANIn3 and ANIn5 to ANIn7 pins do not exceed the rated values. If a voltage higher than or equal to AV_{REFPn} or lower than or equal to AV_{SSn} (even within the range of the absolute maximum ratings) is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

(9) AV_{REFPn} pin (n = 0, 1)

This pin is used for inputting the reference voltage of A/D converters 0 and 1. It converts signals input to the analog input pin to digital signals based on the voltage applied between AV_{REFPn} and AV_{SSn} (n = 0, 1).

Always make the potential at this pin the same as that at the EV_{DD0}, EV_{DD1}, EV_{DD2}, and EV_{DD3} (V850E/IH4-H only) pins even when A/D converters 0 and 1 are not used.

The operating voltage range of the AV_{REFPn} pin is $EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3}$ (V850E/IH4-H only) $= AV_{DDn} = AV_{REFPn} = 4.0$ to 5.5 V.

(10) AV$_{SSn}$ pin (n = 0, 1)

This is the ground pin of A/D converters 0 and 1. Always make the potential at this pin the same as that at the EV$_{SS0}$, EV$_{SS1}$, EV$_{SS2}$, EV$_{SS3}$ (V850E/IH4-H only), and EV$_{SS4}$ pins even when A/D converters 0 and 1 are not used.

(11) AV$_{DDn}$ pin (n = 0, 1)

This pin is the analog power supply pin of A/D converters 0 and 1.

Supply the same potential to the AV$_{DD0}$ and AV$_{DD1}$ pins.

Always make the potential at this pin the same as that at the EV$_{DD0}$, EV$_{DD1}$, EV$_{DD2}$, and EV$_{DD3}$ (V850E/IH4-H only) pins even when A/D converters 0 and 1 are not used.

The operating voltage range of the AV$_{DDn}$ pin is EV$_{DD0}$ = EV$_{DD1}$ = EV$_{DD2}$ = EV$_{DD3}$ (V850E/IH4-H only) = AV$_{REFPn}$ = AV$_{DDn}$ = 4.0 to 5.5 V.

(12) D/A converter n (n = 0, 1)

Two channels are provided for D/A converter n. D/A converter n0 generates the reference voltage supplied to the comparators for low range overvoltage detection, and D/A converter n1 generates the one for full range overvoltage detection (low range reference voltage: 0.2 to 2.4 V, full range reference voltage: 0.2 to 4.5 V).

12.3 Control Registers

A/D converters 0 and 1 are controlled by the following registers.

- A/D converter n scan mode register (ADnSCM)
- A/D converter n scan mode register L (ADnSCML)
- A/D converter n scan mode register H (ADnSCMH)
- A/D converter n conversion time control register (ADnCTC)
- A/D converter n conversion channel specification register (ADnCHEN)
- A/D converter n conversion channel specification register L (ADnCHENL)
- A/D converter n conversion channel specification register H (ADnCHENH)
- A/D converter n control register (ADnCTL0)
- A/D converter n trigger select register (ADnTSEL)
- A/D converter n channel specification registers 1 and 2 (ADnCH1, ADnCH2)
- A/D converter n flag register (ADnFLG)
- A/D converter n flag buffer register (ADnFLGB)
- A/DLDTRG1 input select register (ADLTS1)
- A/DLDTRG2 input select register (ADLTS2)
- A/D converter n clock select register (ADnOCKS)
- A/D trigger falling edge specification register (ADTF)
- A/D trigger rising edge specification register (ADTR)
- Operational amplifier n control register 0 (OPnCTL0)
- Comparator n control registers 0 to 3 (CMPnCTL0 to CMPnCTL3)
- Comparator output digital noise elimination registers nL, nF (CMPNFCnL, CMPNFCnF)
- Comparator output interrupt rising edge specification register (CMPOR)
- Comparator output interrupt falling edge specification register (CMPOF)
- D/A converter n mode register (DAnM)
- D/A converter n conversion value setting register 0, 1 (DAnCS0, DAnCS1)

The following registers are also used.

- A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15)
- A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H)
- A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4)
- A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H)

(1) A/D converter n scan mode register (ADnSCM)

The ADnSCM register is a register that specifies the normal operation mode and controls conversion operations.

This register can be read or written in 16-bit units.

When the higher 8 bits of the ADnSCM register are used as the ADnSCMH register and the lower 8 bits, as the ADnSCML register, these registers can be read or written in 1-bit or 8-bit units. However, bit 14 is read-only.

Reset sets this register to 0000H.

(1/3)

After reset: 0000H R/W Address: AD0SCM FFFFF220H, AD1SCM FFFFF2A0H

	<15>	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnSCM (n = 0, 1)	ADn CE	ADn CS	0	0	0	ADn PLM	ADn TRG1	ADn TRG0	ADn PS	0	0	0	0	0	0[Note 1]	0

Notes 1. When using A/D converters 1 and 0, be sure to set bit 1 to "1".
This setting can be performed at the same time as other ADnSCM register bits.

ADnCE	A/D conversion operation control
0	Stop conversion operation
1	Start conversion operation

ADnCS	Status of A/D converter n[Note 2]
0	A/D conversion stopped
1	A/D conversion operating (remains “1” even when the channel is changed during successive conversion)

ADnPLM	ADnTRG1	ADnTRG0	Normal operation mode specification
0	0	0	A/D trigger mode
0	0	1	Hardware trigger mode[Note 3]
1	0	0	A/D trigger polling mode
Other than above			Setting prohibited

ADnPS	A/D power save mode specification
0	A/D power save mode
1	A/D operational mode

Notes 2. The ADnCS bit is set to 1 five base clocks (f_{AD01}) after the ADnCE bit has been set to 1 and A/D conversion has been started.
A/D conversion is started when a trigger signal, such as one from a timer, is input in the hardware trigger mode, conversion channel specification mode, or extension buffer mode. In the A/D trigger mode and A/D trigger polling mode, it is started when the ADnCE bit is 1.

3. In the extended operation mode (conversion channel specification mode or extension buffer mode), be sure to set the hardware trigger mode.

(2/3)

Cautions 1. In the A/D trigger mode or the A/D trigger polling mode, conversion is triggered when 1 is written to the ADnCE bit.

In the hardware trigger mode, the conversion channel specification mode, or the extension buffer mode, the trigger signal wait state starts when 1 is written to the ADnCE bit.

The ADnCE bit is not cleared to 0 even after the A/Dn conversion end interrupt request signal (INTADn) is generated in all modes. To stop the A/D conversion operation, therefore, write 0 to the ADnCE bit.

2. If the ADnSCM register is written during A/D conversion operation (ADnCS bit = 1), the operation is performed as follows in each mode. The corresponding conversion result register is undefined during A/D conversion operation.

- **In A/D trigger mode, A/D trigger polling mode**
  A/D conversion is stopped and executed again from the beginning.
- **In hardware trigger mode, conversion channel specification mode, extension buffer mode**
  A/D conversion is stopped and the trigger standby state is restored again.

3. Make sure that time of at least five base clocks (f_{AD01}) passes before successively writing data to the ADnSCM register when the conversion operation is enabled (ADnCE bit = 1). Otherwise, the register may not be set correctly.

The register can be successively written if the ADnCE bit is set to 1 after the ADnSCM register is written when ADnCE bit = 0.

4. The ADnCS bit remains set (1) when the conversion channel is changed during successive conversion.

5. It is recommended to set the A/D power save mode (ADnPS bit = 0) when the A/D converter is not used.

(3/3)

Cautions 6. The setting procedure is as follows when an A/D conversion operation is started (after a reset ends and after recovery from the A/D power save mode (ADnPS bit = 0)).

<1> Select an input clock (f_{AD01}) by using the ADnOCKS register and set the ADnOCKSEN bit to 1 (to enable the supply of the operating clock to A/D converter n).

<2> Set the A/D conversion time by using the ADnCTC.ADnFR3 to ADnFR0 bits.

<3> Set the ADnPS bit to 1 (A/D operation mode).

<4> Wait 1 μs or longer.

<5> Set up A/D converters 0 and 1.

<6> Set the ADnCE bit to 1 (to enable conversion).

The setting procedure is as follows when using a comparator.

<1> Set the conversion value for D/A converter ny by using the DAnCSy register (y = 0, 1).

<2> Set the DAnM.DAnCEy bit to 1 (to start D/A conversion).

<3> Wait 10 μs or longer (D/A converter ny settling time).

<4> Set the corresponding bit of the CMPnCTL0 register to 1 (to start comparator operation).

<5> Wait 10 μs or longer (comparator stabilization time).

<6> Set up A/D converters 0 and 1.

To change the reference voltage, clear the corresponding bit of the CMPnCTL0 register to 0 (to stop comparator operation), leave D/A conversion enabled, rewrite the DAnCSy register, and specify the settings again from <3>.

7. The setting procedure is as follows when A/D conversion is stopped.

<1> Clear the ADnCE bit to 0 (to stop conversion) (retaining ADnPS bit = 1).

<2> Clear the ADnPS bit to 0 (A/D power save mode).

<3> Clear the ADnOCKS.ADnOCKSEN bit to 0 (to stop supplying the operating clock to A/D converter n).

The setting procedure is as follows when using a comparator.

<1> Clear the corresponding bit of the CMPnCTL0 register to 0 (to stop comparator operation, retaining DAnM.DAnCEy bit = 1 (y = 0, 1)).

<2> Clear the DAnM.DAnCEy bit to 0 (to stop D/A conversion).

8. It is recommended to set the A/D power save mode even in the IDLE and STOP modes. Follow the setting procedure in Caution 6 above when releasing the IDLE or STOP mode by using the reset signal.

9. Be sure to clear bits 0, 2 to 6 and 11 to 13 to “0”.

(2) A/D converter n conversion time control register (ADnCTC)

The ADnCTC register is a register that specifies the number of A/D conversion clocks and A/D conversion time.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0CTC FFFFF222H, AD1CTC FFFFF2A2H

	7	6	5	4	3	2	1	0
ADnCTC (n = 0, 1)	0	0	0	0	ADnFR3	ADnFR2	ADnFR1	ADnFR0

Cautions 1. See Table 12-2 Number of A/D Conversion Clocks and A/D Conversion Time for the ADnFR3 to ADnFR0 bits.

2. Set the ADnFR3 to ADnFR0 bits when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped).

3. Be sure to set bits 4 to 7 to “0”.

Table 12-2. Number of A/D Conversion Clocks and A/D Conversion Time

ADnFR3	ADnFR2	ADnFR1	ADnFR0	Number of A/D Conversion Clocks[Note 1]	A/D Conversion Time (μs)[Note 2]			
					f_{AD01} = 16.66 MHz	f_{AD01} = 16 MHz	f_{AD01} = 12.5 MHz	f_{AD01} = 10 MHz
0	0	0	0	89	5.34	5.56	7.12	Setting prohibited
0	0	0	1	88	5.28	5.50	7.04	Setting prohibited
0	0	1	0	57	3.42	3.56	4.56	5.70
0	0	1	1	56	3.36	3.50	4.48	5.60
0	1	0	0	41	2.46	2.56	3.28	4.10
0	1	0	1	40	2.40	2.50	3.20	4.00
0	1	1	0	35	2.10	2.19	2.80	3.50
0	1	1	1	34	2.04	2.13	2.72	3.40
1	0	0	0	34	2.04	2.13	2.72	3.40
1	0	0	1	33	Setting prohibited	2.06	2.64	3.30
1	0	1	0	33	Setting prohibited	2.06	2.64	3.30
1	0	1	1	32	Setting prohibited	2.00	2.56	3.20
1	1	0	0	32	Setting prohibited	2.00	2.56	3.20
1	1	0	1	31	Setting prohibited	Setting prohibited	2.48	3.10
1	1	1	0	31	Setting prohibited	Setting prohibited	2.48	3.10
1	1	1	1	30	Setting prohibited	Setting prohibited	2.40	3.00

Notes 1. The number of clocks (f_{AD01}) from the start to the end of A/D conversion.
The number of clocks (f_{AD01}) per conversion during successive conversion (1-channel conversion (repeat), multiple channel conversion, or multiple channel conversion (repeat)) is the same.

2. Set the A/D conversion time in a range of 2.00 to 8.00 μs.
A/D Conversion time = $1/f_{AD01} \times$ Number of A/D conversion clocks

(3) A/D converter n conversion channel specification register (ADnCHEN)

The ADnCHEN register is a register that specifies the analog input pin, number of conversion times, and conversion result register.

This register is used to specify an analog input pin in the A/D trigger mode, A/D trigger polling mode, and hardware trigger mode. The ADnCRm register corresponds to an analog input pin on a one-to-one basis. Use the bits (AD0CHEN00 to AD0CHEN05 and AD1CHEN00 to AD1CHEN07) corresponding to the ANI00 to ANI05 and ANI10 to ANI17 pins. If two or more analog input pins are specified, they are sequentially selected, starting from the one with the lowest number, for conversion (when AD1CHEN register = 004DH: ANI10 → ANI12 → ANI13 → ANI16). If an analog input pin that is not specified is skipped during successive conversion.

In the conversion channel specification mode, specify the number of times of conversion and a conversion result register. Specify an analog input pin by using the ADnCH1 register. A value set to the lower bits of the ADnCHEN register, justified to the lowest bit, is the number of times of conversion. These bits correspond to the ADnCRm and ADnCHmH registers on a one-to-one basis.

Because the ADnCHEN register is of master/slave configuration, a new analog input pin can be set to the master register during A/D conversion operation. The set value of the master register is transferred to a slave register after completion of A/D conversion (after the A/Dn conversion end interrupt request signal (INTADn) is generated).

This register can be read or written in 16-bit units.

When the higher 8 bits of the ADnCHEN register are used as the ADnCHENH register and the lower 8 bits, as the ADnCHENL register, these registers can be read or written in 1-bit or 8-bit units.

Reset sets this register to 0000H.

After reset: 0000H R/W Address: AD0CHEN FFFFF224H, AD1CHEN FFFFF2A4H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnCHEN (n = 0, 1)	ADn CHEN 15	ADn CHEN 14	ADn CHEN 13	ADn CHEN 12	ADn CHEN 11	ADn CHEN 10	ADn CHEN 9	ADn CHEN 8	ADn CHEN 7	ADn CHEN 6	ADn CHEN 5	ADn CHEN 4	ADn CHEN 3	ADn CHEN 2	ADn CHEN 1	ADn CHEN 0

Remark See **Table 12-3 Specifying Analog Input Pin in A/D Trigger Mode, A/D Trigger Polling Mode, and Hardware Trigger Mode** for how to specify an analog input pin in the A/D trigger mode, A/D trigger polling mode, and hardware trigger mode. For how to specify the number of times of conversion and the A/D conversion result register in the conversion channel specification mode, see **Table 12-4 Correspondence Among Set Value of ADnCHEN Register, Number of Times of Conversion, and A/D Conversion Result Register in Conversion Channel Specification Mode**.

Cautions 1. The A/D conversion operation is prohibited when the ADnCHEN register = 0000H.
If the ADnCHEN register = 0000H, the operation is the same as when the ADnCHEN register = 0001H.

2. Do not write the ADnCHEN register when the ADnSCM.ADnPS bit = 0. If it is written, the CPU deadlocks.

3. To change the setting of the ADnCHEN register when the ADnSCM.ADnCE bit = 1 in the hardware trigger mode, be sure to set the ADnCE bit to 0.

Table 12-3. Specifying Analog Input Pin in A/D Trigger Mode, A/D Trigger Polling Mode, and Hardware Trigger Mode

ADnCHENm Bit	Specification of Analog Input Pin
0	Specifying ANInk pin is prohibited.
1	Specifying ANInk pin is enabled.

Remark A/D converter 0: n = 0, k = 0 to 3, 5 to 7, m = 0 to 15
A/D converter 1: n = 1,
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7
m = 0 to 15

Table 12-4. Correspondence Among Set Value of ADnCHEN Register, Number of Times of Conversion, and A/D Conversion Result Register in Conversion Channel Specification Mode

ADnCHEN Register Value	Number of Times of Conversion	A/D Conversion Result Register	
0001H	1	ADnCR0	ADnCR0H
0003H	2	ADnCR0, ADnCR1	ADnCR0H, ADnCR1H
0007H	3	ADnCR0 to ADnCR2	ADnCR0H to ADnCR2H
000FH	4	ADnCR0 to ADnCR3	ADnCR0H to ADnCR3H
001FH	5	ADnCR0 to ADnCR4	ADnCR0H to ADnCR4H
003FH	6	ADnCR0 to ADnCR5	ADnCR0H to ADnCR5H
007FH	7	ADnCR0 to ADnCR6	ADnCR0H to ADnCR6H
00FFH	8	ADnCR0 to ADnCR7	ADnCR0H to ADnCR7H
01FFH	9	ADnCR0 to ADnCR8	ADnCR0H to ADnCR8H
03FFH	10	ADnCR0 to ADnCR9	ADnCR0H to ADnCR9H
07FFH	11	ADnCR0 to ADnCR10	ADnCR0H to ADnCR10H
0FFFH	12	ADnCR0 to ADnCR11	ADnCR0H to ADnCR11H
1FFFH	13	ADnCR0 to ADnCR12	ADnCR0H to ADnCR12H
3FFFH	14	ADnCR0 to ADnCR13	ADnCR0H to ADnCR13H
7FFFH	15	ADnCR0 to ADnCR14	ADnCR0H to ADnCR14H
FFFFH	16	ADnCR0 to ADnCR15	ADnCR0H to ADnCR15H
Others	Setting prohibited		

Caution An analog input pin is specified by the ADnCH1 register in the conversion channel specification mode.

Remark n = 0, 1

(4) A/Dn conversion result registers 0 to 15, 0H to 15H (ADnCR0 to ADnCR15, ADnCR0H to ADnCR15H)

The ADnCRm and ADnCRmH registers are registers that hold the A/D conversion results in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, or conversion channel specification mode. Sixteen of these registers are provided per circuit, and two circuits are available. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 12 bits of the ADnCRm register. The lower 4 bits of these registers are always 0 when read.

The higher 8 bits of A/D conversion result are read to the ADnCRmH register.

These registers can only be read in 16-bit or 8-bit units. When the A/D conversion results are read in 16-bit units, the ADnCRm register is specified, and when the higher 8 bits are read, the ADnCRmH register is specified.

Reset sets these registers to 0000H.

Remark While the result of A/D conversion is stored in the ADnCRm register, a read access to the same register is held pending. The pending read access is executed after the A/D conversion result is stored. Similarly, storing the result of A/D conversion in the ADnCRm register is held pending while a read access to that register is made. The pending A/D conversion result storing processing is executed after completion of the read access.

After reset: 0000H R Address: AD0CR0 FFFFF200H, AD0CR1 FFFFF202H,
AD0CR2 FFFFF204H, AD0CR3 FFFFF206H,
AD0CR4 FFFFF208H, AD0CR5 FFFFF20AH,
AD0CR6 FFFFF20CH, AD0CR7 FFFFF20EH,
AD0CR8 FFFFF210H, AD0CR9 FFFFF212H,
AD0CR10 FFFFF214H, AD0CR11 FFFFF216H,
AD0CR12 FFFFF218H, AD0CR13 FFFFF21AH,
AD0CR14 FFFFF21CH, AD0CR15 FFFFF21EH,
AD1CR0 FFFFF280H, AD1CR1 FFFFF282H,
AD1CR2 FFFFF284H, AD1CR3 FFFFF286H,
AD1CR4 FFFFF288H, AD1CR5 FFFFF28AH,
AD1CR6 FFFFF28CH, AD1CR7 FFFFF28EH,
AD1CR8 FFFFF290H, AD1CR9 FFFFF292H,
AD1CR10 FFFFF294H, AD1CR11 FFFFF296H,
AD1CR12 FFFFF298H, AD1CR13 FFFFF29AH,
AD1CR14 FFFFF29CH, AD1CR15 FFFFF29EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnCRm n = 0, 1 m = 0 to 15	ADn CRm 11	ADn CRm 10	ADn CRm 9	ADn CRm 8	ADn CRm 7	ADn CRm 6	ADn CRm 5	ADn CRm 4	ADn CRm 3	ADn CRm 2	ADn CRm 1	ADn CRm 0	0	0	0	0

After reset: 0000H R Address: AD0CR0H FFFFF201H, AD0CR1H FFFFF203H,
AD0CR2H FFFFF205H, AD0CR3H FFFFF207H,
AD0CR4H FFFFF209H, AD0CR5H FFFFF20BH,
AD0CR6H FFFFF20DH, AD0CR7H FFFFF20FH,
AD0CR8H FFFFF211H, AD0CR9H FFFFF213H,
AD0CR10H FFFFF215H, AD0CR11H FFFFF217H,
AD0CR12H FFFFF219H, AD0CR13H FFFFF21BH,
AD0CR14H FFFFF21DH, AD0CR15H FFFFF21FH,
AD1CR0H FFFFF281H, AD1CR1H FFFFF283H,
AD1CR2H FFFFF285H, AD1CR3H FFFFF287H,
AD1CR4H FFFFF289H, AD1CR5H FFFFF28BH,
AD1CR6H FFFFF28DH, AD1CR7H FFFFF28FH,
AD1CR8H FFFFF291H, AD1CR9H FFFFF293H,
AD1CR10H FFFFF295H, AD1CR11H FFFFF297H,
AD1CR12H FFFFF299H, AD1CR13H FFFFF29BH,
AD1CR14H FFFFF29DH, AD1CR15H FFFFF29FH

	7	6	5	4	3	2	1	0
ADnCRmH n = 0, 1 m = 0 to 15	ADnCRm11	ADnCRm10	ADnCRm9	ADnCRm8	ADnCRm7	ADnCRm6	ADnCRm5	ADnCRm4

The correspondence between the analog input pins and the A/D conversion result registers in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, and conversion channel specification mode is shown below.

Table 12-5. Correspondence Between Analog Input Pins and A/D Conversion Result Registers in A/D Trigger Mode, A/D Trigger Polling Mode, Hardware Trigger Mode

A/D Converter	Analog Input Pin	A/D Conversion Result Register
A/D converter 0	ANI00	AD0CR0, AD0CR0H
	ANI01	AD0CR1, AD0CR1H
	ANI02	AD0CR2, AD0CR2H
	ANI03	AD0CR3, AD0CR3H
	ANI05	AD0CR5, AD0CR5H
	ANI06	AD0CR6, AD0CR6H
	ANI07	AD0CR7, AD0CR7H
A/D converter 1	ANI10	AD1CR0, AD1CR0H
	ANI11	AD1CR1, AD1CR1H
	ANI12	AD1CR2, AD1CR2H
	ANI13[Note]	AD1CR3, AD1CR3H
	ANI15	AD1CR5, AD1CR5H
	ANI16	AD1CR6, AD1CR6H
	ANI17	AD1CR7, AD1CR7H

Note V850E/IH4-H only

Table 12-6. Correspondence Between Analog Input Pins and A/D Conversion Result Registers in Conversion Channel Specification Mode

ADnCHEN Register Set Value	Analog Input Pin	A/D Conversion Result Register	
0001H	Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits	ADnCR0	ADnCR0H
0003H		ADnCR0, ADnCR1	ADnCR0H, ADnCR1H
0007H		ADnCR0 to ADnCR2	ADnCR0H to ADnCR2H
000FH		ADnCR0 to ADnCR3	ADnCR0H to ADnCR3H
001FH		ADnCR0 to ADnCR4	ADnCR0H to ADnCR4H
003FH		ADnCR0 to ADnCR5	ADnCR0H to ADnCR5H
007FH		ADnCR0 to ADnCR6	ADnCR0H to ADnCR6H
00FFH		ADnCR0 to ADnCR7	ADnCR0H to ADnCR7H
01FFH		ADnCR0 to ADnCR8	ADnCR0H to ADnCR8H
03FFH		ADnCR0 to ADnCR9	ADnCR0H to ADnCR9H
07FFH		ADnCR0 to ADnCR10	ADnCR0H to ADnCR10H
0FFFH		ADnCR0 to ADnCR11	ADnCR0H to ADnCR11H
1FFFH		ADnCR0 to ADnCR12	ADnCR0H to ADnCR12H
3FFFH		ADnCR0 to ADnCR13	ADnCR0H to ADnCR13H
7FFFH		ADnCR0 to ADnCR14	ADnCR0H to ADnCR14H
FFFFH		ADnCR0 to ADnCR15	ADnCR0H to ADnCR15H
0001H	Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits	ADnCR0	ADnCR0H
0003H		ADnCR0, ADnCR1	ADnCR0H, ADnCR1H
0007H		ADnCR0 to ADnCR2	ADnCR0H to ADnCR2H
000FH		ADnCR0 to ADnCR3	ADnCR0H to ADnCR3H
001FH		ADnCR0 to ADnCR4	ADnCR0H to ADnCR4H
003FH		ADnCR0 to ADnCR5	ADnCR0H to ADnCR5H
007FH		ADnCR0 to ADnCR6	ADnCR0H to ADnCR6H
00FFH		ADnCR0 to ADnCR7	ADnCR0H to ADnCR7H
01FFH		ADnCR0 to ADnCR8	ADnCR0H to ADnCR8H
03FFH		ADnCR0 to ADnCR9	ADnCR0H to ADnCR9H
07FFH		ADnCR0 to ADnCR10	ADnCR0H to ADnCR10H
0FFFH		ADnCR0 to ADnCR11	ADnCR0H to ADnCR11H
1FFFH		ADnCR0 to ADnCR12	ADnCR0H to ADnCR12H
3FFFH		ADnCR0 to ADnCR13	ADnCR0H to ADnCR13H
7FFFH		ADnCR0 to ADnCR14	ADnCR0H to ADnCR14H
FFFFH		ADnCR0 to ADnCR15	ADnCR0H to ADnCR15H
Others	Setting prohibited		

Remark n = 0, 1

(5) A/D converter n control register (ADnCTL0)

The ADnCTL0 register is a register that specifies the operation mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0CTL0 FFFFF230H, AD1CTL0 FFFFF2B0H

	7	6	5	4	3	2	1	0
ADnCTL0 (n = 0, 1)	0	0	0	0	0	0	ADnMD1	ADnMD0

ADnMD1	ADnMD0	Extended operating mode specification
0	0	Normal operating mode
0	1	Setting prohibited
1	0	Conversion channel specification mode
1	1	Extension buffer mode

Cautions 1. Set the ADnMD1 and ADnMD0 bits when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to these bits when the ADnCE bit = 1 (conversion operation is enabled)).

2. In the conversion channel specification mode and extension buffer mode, start of A/D conversion is delayed up to 1.5 base clocks (f_{AD01}) as compared with the normal operating mode.

3. Be sure to set the hardware trigger mode in the conversion channel specification mode and extension buffer mode.

(6) A/D converter n trigger select register (ADnTSEL)

The ADnTSEL register is a register that specifies trigger in the hardware trigger mode and conversion channel specification mode, and trigger (selection trigger 1, selection trigger 2, selection load trigger 1, and selection load trigger 2) in the extension buffer mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 10H.

After reset: 10H R/W Address: AD0TSEL FFFFF231H, AD1TSEL FFFFF2B1H

	7	6	5	4	3	2	1	0
ADnTSEL (n = 0, 1)	ADn Note LDTSEL2	0	ADn Note TRGSEL21	ADn Note TRGSEL20	ADn Note LDTSEL1	0	ADn TRGSEL11	ADn TRGSEL10

ADnLDTSEL2 Note	Specification of selection load trigger 2 for ADnECR3, ADnECR4 registers
0	LDTRG1
1	LDTRG2

ADnTRGSEL21 Note	ADnTRGSEL20 Note	Specification of selection trigger 2 for ADnECR3, ADnECR4 registers
0	0	ITRG1
0	1	ITRG2
1	0	ITRG3
1	1	ITRG4

ADnLDTSEL1 Note	Specification of selection load trigger 1 for ADnECR0 to ADnECR2 registers
0	LDTRG1
1	LDTRG2

ADnTRGSEL11	ADnTRGSEL10	• In hardware trigger mode or conversion channel specification mode: Trigger specification • In expansion buffer mode: Specification of selection trigger 1 for ADnECR0 to ADnECR2 registers
0	0	ITRG1
0	1	ITRG2
1	0	ITRG3
1	1	ITRG4

Note Be sure to set bits 3, 5, and 7 to "0" and set bit 4 to "1" in the hardware trigger mode and conversion channel specification mode.

Caution Set the ADnTSEL register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).

(7) A/D converter n channel specification register 1 (ADnCH1)

The ADnCH1 register is a register that specifies the analog input pin for selection trigger 1 in the conversion channel specification mode and extension buffer mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0CH1 FFFFF232H, AD1CH1 FFFFF2B2H

	7	6	5	4	3	2	1	0
ADnCH1 (n = 0, 1)	0	ADn TRGCH16	ADn TRGCH15	ADn TRGCH14	0	ADn TRGCH12	ADn TRGCH11	ADn TRGCH10

ADnTRGCH16	ADnTRGCH15	ADnTRGCH14	Specification of analog input pin for selection trigger 1
0	0	0	ANIn0
0	0	1	ANIn1
0	1	0	ANIn2
0	1	1	ANIn3[Note]
1	0	0	Setting prohibited
1	0	1	ANIn5
1	1	0	ANIn6
1	1	1	ANIn7

ADnTRGCH12	ADnTRGCH11	ADnTRGCH10	Specification of analog input pin for selection trigger 1
0	0	0	ANIn0
0	0	1	ANIn1
0	1	0	ANIn2
0	1	1	ANIn3[Note]
1	0	0	Setting prohibited
1	0	1	ANIn5
1	1	0	ANIn6
1	1	1	ANIn7

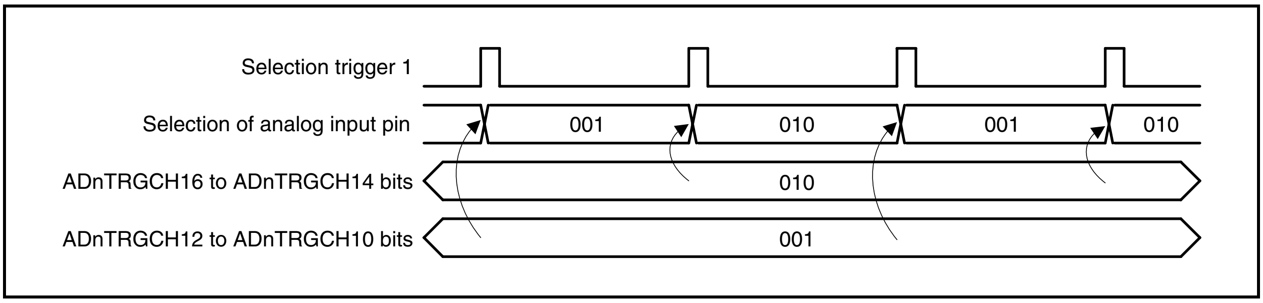
Note For the V850E/IG4-H, this can be specified only for A/D converter 0.
Specifying this for A/D converter 1 is prohibited.

Cautions 1. Set the ADnCH1 register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).

2. Be sure to set bits 3 and 7 to “0”.

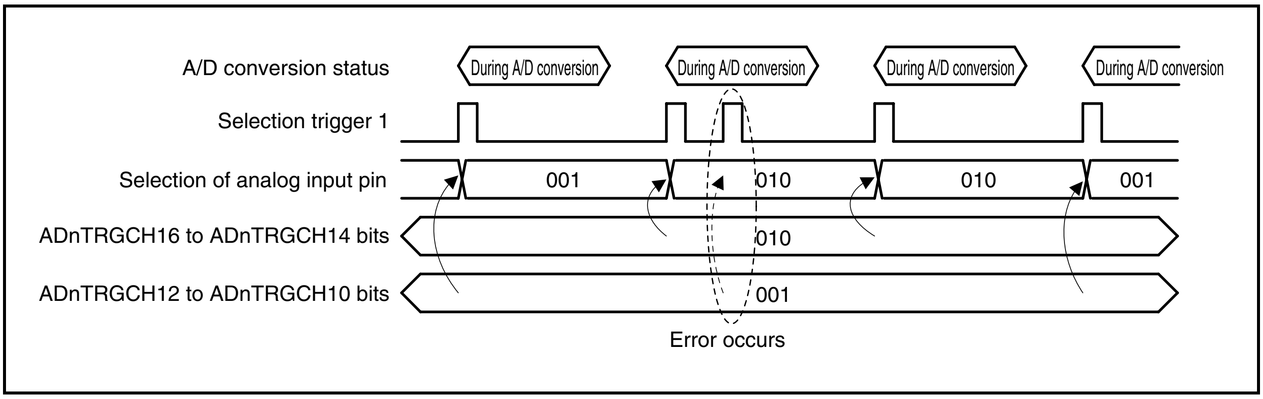
Setting the ADnCH1 register is enabled when a conversion operation is enabled (ADnSCM.ADnCE bit = 1) in the conversion channel specification mode or extension buffer mode. When the first selection trigger 1 is generated after the conversion operation is enabled (ADnCE bit = 1), the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits is selected and A/D conversion is executed. When the next selection trigger 1 is later generated, the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits is selected and A/D conversion is executed. After that, the analog input pins are alternately selected for output each time selection trigger 1 is generated.

Figure 12-7. ADnCH1 Register Operation


Selection trigger 1
Selection of analog input pin
001
010
001
010
ADnTRGCH16 to ADnTRGCH14 bits
010
ADnTRGCH12 to ADnTRGCH10 bits
001

If an error occurs (when selection trigger 1 is generated during A/D conversion), the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits and the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits are alternately selected, but the selected analog input pin is not changed because A/D conversion is in progress.

Figure 12-8. ADnCH1 Register Operation In Case of Error



(8) A/D converter n channel specification register 2 (ADnCH2)

The ADnCH2 register is a register that specifies the analog input pin for selection trigger 2 in the extension buffer mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0CH2 FFFFF233H, AD1CH2 FFFFF2B3H

	7	6	5	4	3	2	1	0
ADnCH2 (n = 0, 1)	0	ADn TRGCH26	ADn TRGCH25	ADn TRGCH24	0	ADn TRGCH22	ADn TRGCH21	ADn TRGCH20

ADnTRGCH26	ADnTRGCH25	ADnTRGCH24	Specification of analog input pin for selection trigger 2
0	0	0	ANIn0
0	0	1	ANIn1
0	1	0	ANIn2
0	1	1	ANIn3^Note
1	0	0	Setting prohibited
1	0	1	ANIn5
1	1	0	ANIn6
1	1	1	ANIn7

ADnTRGCH22	ADnTRGCH21	ADnTRGCH20	Specification of analog input pin for selection trigger 2
0	0	0	ANIn0
0	0	1	ANIn1
0	1	0	ANIn2
0	1	1	ANIn3^Note
1	0	0	Setting prohibited
1	0	1	ANIn5
1	1	0	ANIn6
1	1	1	ANIn7

Note For the V850E/IG4-H, this can be specified only for A/D converter 0.
Specifying this for A/D converter 1 is prohibited.

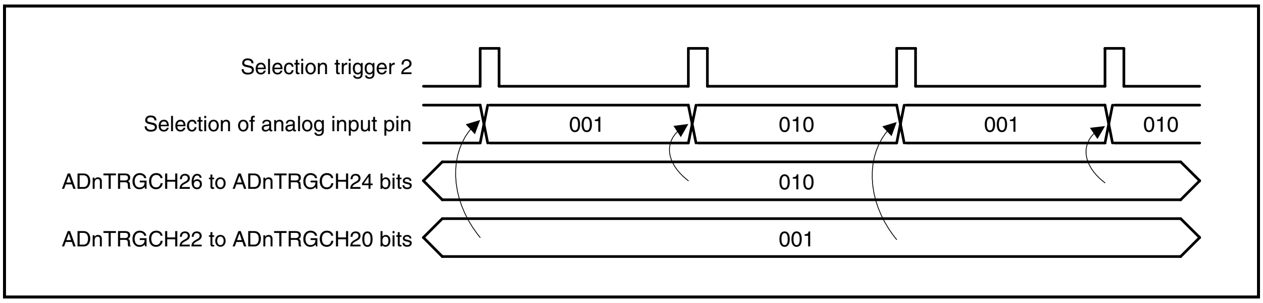
Cautions 1. Set the ADnCH2 register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).

2. The ADnCH2 register is valid only in the extension buffer mode; it is invalid in any other mode.

3. Be sure to set bits 3 and 7 to “0”.

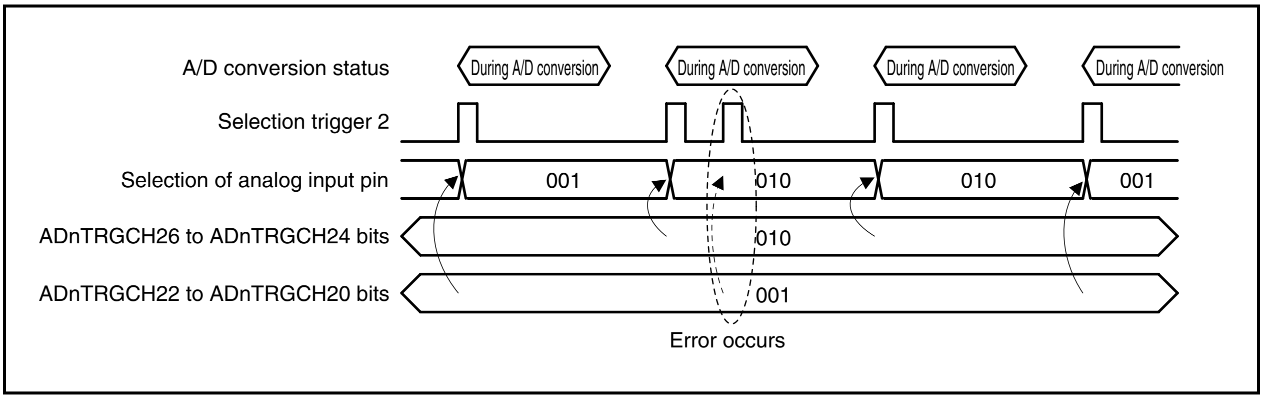
Setting the ADnCH2 register is enabled when a conversion operation is enabled (ADnSCM.ADnCE bit = 1) in the extension buffer mode. When the first selection trigger 2 is generated after the conversion operation is enabled (ADnCE bit = 1), the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits is selected and A/D conversion is executed. When the next selection trigger 2 is later generated, the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits is selected and A/D conversion is executed. After that, the analog input pins are alternately selected for output each time selection trigger 2 is generated.

Figure 12-9. ADnCH2 Register Operation


Selection trigger 2
Selection of analog input pin
001
010
001
010
ADnTRGCH26 to ADnTRGCH24 bits
010
ADnTRGCH22 to ADnTRGCH20 bits
001

If an error occurs (when selection trigger 2 is generated during A/D conversion), the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits and the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits are alternately selected, but the selected analog input pin is not changed because A/D conversion is in progress.

Figure 12-10. ADnCH2 Register Operation In Case of Error



(9) A/Dn conversion result extension registers 0 to 4, 0H to 4H (ADnECR0 to ADnECR4, ADnECR0H to ADnECR4H)

The ADnECRa and ADnECRaH registers hold the result of A/D conversion in their higher 12 bits and indicate the status (information on the A/D conversion result of the analog input pin specified by the ADnCHx.ADnTRGCHx2 to ADnTRGCHx0 bits or ADnTRGCHx6 to ADnTRGCHx4 bits) of the A/D conversion result with the lower 1 bit in the extension buffer mode. Five of these registers are provided per circuit and two circuits are available. When A/D conversion is completed, the A/D conversion result is stored in A/Dn conversion result extension buffer register a. When selection load trigger 1 is later generated, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 0 to 2 to the higher 12 bits of the ADnECR0 to ADnECR2 registers and stored. Bits 1 to 3 are always 0 when read. When selection load trigger 2 is generated, the A/D conversion result is shifted from the A/Dn conversion result extension buffer registers 3 and 4 to the higher 12 bits of the ADnECR3 and ADnECR4 registers and stored. Bits 1 to 3 are always 0 when read.

The higher 8 bits of the A/D conversion result are read from the ADnECRaH register.

These registers are read-only in 16-bit or 8-bit units. To read the A/D conversion result in 16-bit units, specify the ADnECRa register. Specify the ADnECRaH register to read the higher 8 bits of the A/D conversion result.

Reset sets these registers to 0000H.

Remark While the result of A/D conversion is stored in the ADnECRa register, a read access to that register is held pending. The pending read access is executed when storing the A/D conversion result is completed. Similarly, storing the A/D conversion result in the ADnECRa register is held pending while a read access is made to that register. The pending A/D conversion result is stored in the register after the read access is completed.

After reset: 0000H R Address: AD0ECR0 FFFFF240H, AD0ECR1 FFFFF242H,
AD0ECR2 FFFFF244H, AD0ECR3 FFFFF246H,
AD0ECR4 FFFFF248H,
AD1ECR0 FFFFF2C0H, AD1ECR1 FFFFF2C2H,
AD1ECR2 FFFFF2C4H, AD1ECR3 FFFFF2C6H,
AD1ECR4 FFFFF2C8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnECRa (n = 0, 1 a = 0 to 4)	ADn ECRa 11	ADn ECRa 10	ADn ECRa 9	ADn ECRa 8	ADn ECRa 7	ADn ECRa 6	ADn ECRa 5	ADn ECRa 4	ADn ECRa 3	ADn ECRa 2	ADn ECRa 1	ADn ECRa 0	0	0	0	ADn CH FLGa

ADnCHFLGa	Status of A/D conversion result (x = 1, 2)
0	A/D conversion result for analog input pin set by ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 bits
1	A/D conversion result for analog input pin set by ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits

After reset: 00H R Address: AD0ECR0H FFFFF241H, AD0ECR1H FFFFF243H,
AD0ECR2H FFFFF245H, AD0ECR3H FFFFF247H,
AD0ECR4H FFFFF249H,
AD1ECR0H FFFFF2C1H, AD1ECR1H FFFFF2C3H,
AD1ECR2H FFFFF2C5H, AD1ECR3H FFFFF2C7H,
AD1ECR4H FFFFF2C9H

	7	6	5	4	3	2	1	0
ADnECRaH (n = 0, 1 a = 0 to 4)	ADnECRa11	ADnECRa10	ADnECRa9	ADnECRa8	ADnECRa7	ADnECRa6	ADnECRa5	ADnECRa4

Caution The ADnECRa and ADnECRaH registers are valid only in the extension buffer mode; they are invalid in any other mode.

The correspondence between the analog input pins and the A/Dn conversion result extension registers is shown below.

Table 12-7. Correspondence Between Analog Input Pins and A/D Conversion Result Extension Registers

Analog Input Pin	A/Dn Conversion Result Register
Set with ADnCH1 register's ADnTRGCH12 to ADnTRGCH10, ADnTRGCH16 to ADnTRGCH14 bits	ADnECR0, ADnECR0H
	ADnECR1, ADnECR1H
	ADnECR2, ADnECR2H
Set with ADnCH2 register's ADnTRGCH22 to ADnTRGCH20, ADnTRGCH26 to ADnTRGCH24 bits	ADnECR3, ADnECR3H
	ADnECR4, ADnECR4H

Remark n = 0, 1

(10) A/D converter n flag register (ADnFLG)

The ADnFLG register indicates that an error has occurred when selection load trigger x is generated in the extension buffer mode (x = 1 or 2). The ADnTERR2 and ADnTERR1 flags can only be read and cleared when the conversion operation is stopped (ADnSCM.ADnCE bit = 0).

This register is read-only in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R Address: AD0FLG FFFFF254H, AD1FLG FFFFF2D4H

	7	6	5	4	3	2	1	0
ADnFLG (n = 0, 1)	0	0	0	0	0	0	ADn TERR2[Note]	ADn TERR1[Note]

ADnTERR2[Note]	Occurrence timing error flag of selection load trigger 2
0	Occurrence timing error of selection load trigger 2 has not occurred
1	Occurrence timing error of selection load trigger 2 has occurred

ADnTERR1[Note]	Occurrence timing error flag of selection load trigger 1
0	Occurrence timing error of selection load trigger 1 has not occurred
1	Occurrence timing error of selection load trigger 1 has occurred

Note The ADnTERR2 and ADnTERR1 flags are valid only in the extension buffer mode; they are fixed to 0 in any other mode.

(11) A/D converter n flag buffer register (ADnFLGB)

The ADnFLGB register indicates that an error has occurred when selection trigger x is generated in the extension buffer mode (x = 1 or 2). The ADnTERRB2 and ADnTERRB1 flags can only be read and cleared when the conversion operation is stopped (ADnSCM.ADnCE bit = 0).

This register is read-only in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R Address: AD0FLGB FFFFF255H, AD1FLGB FFFFF2D5H

	7	6	5	4	3	2	1	0
ADnFLGB (n = 0, 1)	0	0	0	0	0	0	ADn TERRB2[Note]	ADn TERRB1[Note]

ADnTERRB2[Note]	Occurrence timing error flag of selection trigger 2
0	Occurrence timing error of selection trigger 2 has not occurred
1	Occurrence timing error of selection trigger 2 has occurred

ADnTERRB1[Note]	Occurrence timing error flag of selection trigger 1
0	Occurrence timing error of selection trigger 1 has not occurred
1	Occurrence timing error of selection trigger 1 has occurred

Note The ADnTERRB2 and ADnTERRB1 flags are valid only in the extension buffer mode; they are fixed to 0 in any other mode.

(12) A/D LDTRG1 input select register (ADLTS1)

The ADLTS1 register is a register that specifies the input signal for selection load trigger (LDTRG1) in the extension buffer mode.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF2F8H

	7	6	5	4	3	2	1	0
ADLTS1	0	0	0	0	0	0	0	ADLTS10

ADLTS10	Specification of input signal for LDTRG1
0	TABTIOV0 signal
1	TABTIOV1 signal

Note The ADLTS1 register is valid only in the extension buffer mode; it is invalid in any other mode.

(13) A/D LDTRG2 input select register (ADLTS2)

The ADLTS2 register is a register that specifies the input signal for selection load trigger (LDTRG2) in the extension buffer mode.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF2FAH

	7	6	5	4	3	2	1	0
ADLTS2	0	0	0	0	0	0	0	ADLTS20

ADLTS20	Specification of input signal for LDTRG2
0	TABTICC00 signal
1	TABTICC10 signal

Note The ADLTS2 register is valid only in the extension buffer mode; it is invalid in any other mode.

(14) A/D converter n clock select register (ADnOCKS)

The ADnOCKS register is a register that selects the clock (f_{AD01}) to be input to the A/D converter n.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0OCKS FFFFF270H, AD1OCKS FFFFF274H

	7	6	5	4	3	2	1	0
ADnOCKS (n = 0, 1)	0	0	0	ADnOCKSEN	0	0	ADnOCKS1	ADnOCKS0

ADnOCKSEN	Clock operation control
0	Stop operation clock supply of A/D converter n
1	Enable operation clock supply of A/D converter n

ADnOCKS1	ADnOCKS0	Input clock selection of A/D converter n (f_{AD01})
0	0	$f_{xx}/4$
0	1	$f_{xx}/6$
1	0	$f_{xx}/8$
1	1	$f_{xx}/10$

Cautions 1. Set f_{AD01} to 4 to 16.7 MHz.

2. When A/D converter n is used, be sure to set the ADnOCKS register and set the ADnSCM.ADnPS bit to 1, as well as to read the A/D conversion result register.

3. Be sure to set bits 2, 3, and 5 to 7 to “0”.

(15) A/D trigger rising edge, falling edge specification registers (ADTR, ADTF)

The ADTR and ADTF registers are registers that specify the trigger mode of the ADTRG0/INTADT0 and ADTRG1/INTADT1 pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

Caution When the function is changed from the external trigger input of the A/D converter n (alternate function)/external interrupt function (alternate function) to the port mode, an edge may be detected. Therefore, be sure to set the ADTFn and ADTRn bits to 00, and then set the port mode.

After reset: 00H R/W Address: FFFFF2F2H

	7	6	5	4	3	2	1	0
ADTR	0	0	0	0	0	0	ADTR1	ADTR0

After reset: 00H R/W Address: FFFFF2F0H

	7	6	5	4	3	2	1	0
ADTF	0	0	0	0	0	0	ADTF1	ADTF0

Remark For the valid edge specification, see **Table 12-8**.

Table 12-8. Valid Edge Specification of ADTRG0/INTADT0 and ADTRG1/INTADT1 Pins

ADTFn	ADTRn	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution When not using these pins as the ADTRGn/INTADTn pins, be sure to set the ADTFn and ADTRn bits to 00.

Remark n = 0, 1

(16) Operational amplifier n control register 0 (OPnCTL0)

The OPnCTL0 register is used to control the operation of an operational amplifier that amplifies the input level, and specify its gain.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: OP0CTL0 FFFFF260H, OP1CTL0 FFFFF2E0H

	7	6	5	4	3	2	1	0
OPnCTL0 (n = 0, 1)	0	OPn2EN	OPn1EN	OPn0EN	OPnGA3	OPnGA2	OPnGA1	OPnGA0

OPn2EN	Operation control of operational amplifier 2 for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

OPn1EN	Operation control of operational amplifier 1 for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

OPn0EN	Operation control of operational amplifier 0 for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

OPnGA3	OPnGA2	OPnGA1	OPnGA0	Gain specification of operational amplifier
0	0	0	0	×2.500
0	0	0	1	×2.667
0	0	1	0	×2.857
0	0	1	1	×3.077
0	1	0	0	×3.333
0	1	0	1	×3.636
0	1	1	0	×4.000
0	1	1	1	×4.444
1	0	0	0	×5.000
1	0	0	1	×5.714
1	0	1	0	×6.667
1	0	1	1	×8.000
1	1	0	0	×10.00
Others				Setting prohibited

Caution After enabling the operational amplifier, a stabilization time of 10 μs is required.
If the settings for the OPnGA3 to OPnGA0 bits have been changed, a stabilization time of 5 μs is required.

(17) Comparator n control register 0 (CMPnCTL0)

The CMPnCTL0 register is a register that controls the operation of the overvoltage detection comparator.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: CMP0CTL0 FFFFF261H, CMP1CTL0 FFFFF2E1H

	7	6	5	4	3	2	1	0
CMPnCTL0 (n = 0, 1)	0	CMPn2FEN	CMPn1FEN	CMPn0FEN	0	CMPn2LEN	CMPn1LEN	CMPn0LEN

CMPn2FEN	Operation control of comparator 2 (full range) for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

CMPn1FEN	Operation control of comparator 1 (full range) for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

CMPn0FEN	Operation control of comparator 0 (full range) for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

CMPn2LEN	Operation control of comparator 2 (low range) for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

CMPn1LEN	Operation control of comparator 1 (low range) for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

CMPn0LEN	Operation control of comparator 0 (low range) for A/D converter n
0	Operation disabled (not used)
1	Operation enabled (used)

Cautions 1. After enabling the operation of the comparator, stabilization time of 10 μs is required.

2. The reference voltages supplied to the comparators are generated by D/A converter n. The reference voltages are in the range below regardless of whether the input signals are amplified by the operational amplifiers.

Low range reference voltage: 0.2 to 2.4 V
Full range reference voltage: 0.2 to 4.5 V

For details, see CHAPTER 28 ELECTRICAL SPECIFICATIONS.

(18) Comparator n control register 1 (CMPnCTL1)

The CMPnCTL1 register is a register that monitors the output of the overvoltage detection comparator.

This register is read-only in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R Address: CM0CTL1 FFFFF262H, CMP1CTL1 FFFFF2E2H

	7	6	5	4	3	2	1	0
CMPnCTL1 (n = 0, 1)	0	CMPn2FOUT	CMPn1FOUT	CMPn0FOUT	0	CMPn2LOUT	CMPn1LOUT	CMPn0LOUT

CMPn2FOUT	Output level status of comparator 2 (full range) for A/D converter n
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMPn1FOUT	Output level status of comparator 1 (full range) for A/D converter n
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMPn0FOUT	Output level status of comparator 0 (full range) for A/D converter n
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMPn2LOUT	Output level status of comparator 2 (low range) for A/D converter n
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMPn1LOUT	Output level status of comparator 1 (low range) for A/D converter n
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMPn0LOUT	Output level status of comparator 0 (low range) for A/D converter n
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

Caution The CMPn2FOUT, CMPn1FOUT, CMPn0FOUT, CMPn2LOUT, CMPn1LOUT, and CMPn0LOUT bits are set to 0 when the input voltage falls to a level at which an overvoltage is not detected.

(19) Comparator n control register 2 (CMPnCTL2)

The CMPnCTL2 register is a register that specifies the compare signal of the overvoltage detection comparator.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: CMP0CTL2 FFFFF263H, CMP1CTL2 FFFFF2E3H

	7	6	5	4	3	2	1	0
CMPnCTL2 (n = 0, 1)	0	0	0	0	0	CMPn2SEL	CMPn1SEL	CMPn0SEL

CMPn2SEL	Specification of compare signal of comparator 2 for A/D converter n
0	Before operational amplifier 2 amplification
1	After operational amplifier 2 amplification

CMPn1SEL	Specification of compare signal of comparator 1 for A/D converter n
0	Before operational amplifier 1 amplification
1	After operational amplifier 1 amplification

CMPn0SEL	Specification of compare signal of comparator 0 for A/D converter n
0	Before operational amplifier 0 amplification
1	After operational amplifier 0 amplification

(20) Comparator n control register 3 (CMPnCTL3)

The CMPnCTL3 register is a register that specifies the detection direction of the overvoltage detection comparator and selects the edge detection.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: CMP0CTL3 FFFFF264H, CMP1CTL3 FFFFF2E4H

	7	6	5	4	3	2	1	0
CMPnCTL3 (n = 0, 1)	CMPnFDS	CMPn2FDE	CMPn1FDE	CMPn0FDE	CMPnLDS	CMPn2LDE	CMPn1LDE	CMPn0LDE

CMPnFDS	Specification of detection direction for comparator (full range) for A/D converter n
0	Logical product (AND) detection (detects whether the input voltage is lower than the reference value)
1	Logical sum (OR) detection (detects whether the input voltage is more than the reference value)

CMPn2FDE	Selection of edge detection for comparator 2 (full range) for A/D converter n
0	Edge detection disabled (comparator not used)
1	Edge detection enabled (comparator used)

CMPn1FDE	Selection of edge detection for comparator 1 (full range) for A/D converter n
0	Edge detection disabled (comparator not used)
1	Edge detection enabled (comparator used)

CMPn0FDE	Selection of edge detection for comparator 0 (full range) for A/D converter n
0	Edge detection disabled (comparator not used)
1	Edge detection enabled (comparator used)

CMPnLDS	Specification of detection direction for comparator (low range) for A/D converter n
0	Logical product (AND) detection (detects whether the input voltage is lower than the reference value)
1	Logical sum (OR) detection (detects whether the input voltage is more than the reference value)

CMPn2LDE	Selection of edge detection for comparator 2 (low range) for A/D converter n
0	Edge detection disabled (comparator not used)
1	Edge detection enabled (comparator used)

CMPn1LDE	Selection of edge detection for comparator 1 (low range) for A/D converter n
0	Edge detection disabled (comparator not used)
1	Edge detection enabled (comparator used)

CMPn0LDE	Selection of edge detection for comparator 0 (low range) for A/D converter n
0	Edge detection disabled (comparator not used)
1	Edge detection enabled (comparator used)

Remark The reference value indicates the reference voltage generated by D/A converter n (n = 0, 1).

(21) Comparator output digital noise elimination register nL, nF (CMPNFCnL, CMPNFCnF)

The CMPNFCnL and CMPNFCnF registers are control the digital noise elimination of the overvoltage detection comparator output.

This register can be read or written in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: CMPNFC0L FFFFF278H, CMPNFC1L FFFFF27CH

	7	6	5	4	3	2	1	0
CMPNFCnL (n = 0, 1)	CMPnNFEN	0	0	0	0	CMPnNFC2	CMPnNFC1	CMPnNFC0

After reset: 00H R/W Address: CMPNFC0F FFFFF27AH, CMPNFC1F FFFFF27EH

	7	6	5	4	3	2	1	0
CMPNFCnF (n = 0, 1)	CMPnNFEN	0	0	0	0	CMPnNFC2	CMPnNFC1	CMPnNFC0

CMPnNFEN	Setting of digital noise elimination
0	Perform analog noise elimination
1	Perform digital noise elimination

CMPnNFC2	CMPnNFC1	CMPnNFC0	Sampling clock selection
0	0	0	fxx/32
0	0	1	fxx/64
0	1	0	fxx/128
0	1	1	fxx/256
1	0	0	fxx/512
1	0	1	fxx/1024
Others			Setting prohibited

Caution Be sure to set bits 3 to 6 to “0”.

(22) Comparator output interrupt rising edge, falling edge specification registers (CMPOR, CMPOF)

The CMPOR and CMPOF registers are registers that specify the trigger mode of the INTCMP0L, INTCMP0F, INTCMP1L, and INTCMP1F signals and can specify the valid edge independently for each interrupt request signal (rising edge, falling edge, or both rising and falling edges).

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

After reset: 00H R/W Address: FFFFF2F6H

	7	6	5	4	3	2	1	0
CMPOR	0	0	0	0	CMPOR1F	CMPOR1L	CMPOR0F	CMPOR0L

After reset: 00H R/W Address: FFFFF2F4H

	7	6	5	4	3	2	1	0
CMPOF	0	0	0	0	CMPOF1F	CMPOF1L	CMPOF0F	CMPOF0L

Remark For the valid edge specification, see **Tables 12-9** and **12-10**.

Table 12-9. Valid Edge Specification of INTCMP0F and INTCMP1F Signals

CMPOFnF	CMPORnF	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 0, 1

Table 12-10. Valid Edge Specification of INTCMP0L and INTCMP1L Signals

CMPOFnL	CMPORnL	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 0, 1

(23) D/A converter n mode register (DAnM)

The DAnM register controls the operation of the D/A converter n.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: DA0M FFFFFB02H, DA1M FFFFFB12H

	7	6	5	4	3	2	1	0
DAnM (n = 0, 1)	0	0	DAnCE1	DAnCE0	0	0	0	0

DAnCE1	D/A converter n1 operation enable/disable
0	Disable operation
1	Enable operation

DAnCE0	D/A converter n0 operation enable/disable
0	Disable operation
1	Enable operation

Caution Be sure to set bits 0 to 3, 6, and 7 to “0”.

(a) D/A converter n operation

D/A conversion is performed using a write operation to the DAnCSy register as the trigger.

The setting method is described below.

<1> Set the analog voltage to be output as the reference voltage of comparator n to the DAnCSy register as the initial settings.

<2> Set the DAnM.DAnCEy bit to 1 (D/A conversion enable).
D/A conversion starts when this setting is performed.

<3> To perform subsequent D/A conversions, write to the DAnCSy register.
The previous D/A conversion result is held until the next D/A conversion is performed.

Remarks 1. For the alternate-function pin settings, see **Table 4-16 Settings When Pins Are Used for Alternate Functions**.

2. n = 0, 1
y = 0, 1

(24) D/A converter conversion n value setting registers 0, 1 (DAnCS0, DAnCS1)

The DAnCS0 and DAnCS1 registers set the analog voltage to be output as the reference voltage of comparator n.

These registers can be read or written in 8-bit units.

Reset sets these registers to 00H.

After reset: 00H R/W Address: DA0CS0 FFFFFB00H, DA0CS1 FFFFFB01H,
DA1CS0 FFFFFB10H, DA1CS1 FFFFFB11H

	7	6	5	4	3	2	1	0
DAnCSy	DAnCSy7	DAnCSy6	DAnCSy5	DAnCSy4	DAnCSy3	DAnCSy2	DAnCSy1	DAnCSy0

n = 0, 1
y = 0, 1

12.4 Operation

Cautions 1. A/D converters 0 and 1 are capable of simultaneous sampling of two circuits.

2. For details of operation setting, see 12.3 (1) A/D converter n scan mode register (ADnSCM).

12.4.1 Basic operation

A/D conversion is executed by the following procedure.

(1) Select an input clock (f_{AD01}) by using the ADnOCKS register and set the ADnOCKSEN bit to 1 (enable supply of the operating clock to A/D converter n).

(2) Set ADnSCM.ADnPS bit = 1.

(3) Wait for 1 μs or more after <2>.

(4) Select an analog input pin and operation mode, by using the ADnSCM[Note], ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, and ADLTS2 registers (n = 0, 1). Number of A/D conversion clocks and A/D conversion time are determined by the specification of the ADnCTC.ADnFR3 to ADnCTC.ADnFR0 bits.

Note Be sure to set bit 1 of the ADnSCM register to “1”.
This setting can be performed at the same time as other ADnSCM register bits.

(5) In the A/D trigger mode and the A/D trigger polling mode, setting the ADnSCM.ADnCE bit to 1 starts A/D conversion (n = 0, 1). If the ADnCE bit is set to 1 in the hardware trigger mode, conversion channel specification mode, and extension buffer mode, the A/D converter enters the trigger wait status.

(6) When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit. When the operational amplifier for input level amplification is used, the gain specified by the OPnCTL0.OPnGA3 to OPnCTL0.OPnGA0 bits × the input voltage is sampled.

(7) To use comparators for overvoltage detection, set up the CMPnCTL0 to CMPnCTL3, CMPNFCnL, CMPNFCnF, CMPOR, CMPOF, DAnM, DAnCS0, and DAnCS1 registers.

(8) When sampling has been performed for a specific time, the sample & hold circuit enters the hold status, and holds the input analog voltage until A/D conversion ends.

(9) Set bit 11 of the successive approximation register (SAR). The tap selector changes the level of the voltage tap of the array to the reference voltage (1/2AV_{REFPn}).

(10) The voltage generated by the voltage tap of the array is compared with the analog input voltage by a comparator. If the analog input voltage is found to be greater than the reference voltage (1/2AV_{REFPn}) as a result of comparison, the most significant bit (MSB) of the successive approximation register (SAR) remains set. If the analog input voltage is less than the reference voltage (1/2AV_{REFPn}), the MSB of the SAR is reset.

(11) Next, bit 10 of the successive approximation register (SAR) is automatically set, and the next comparison is started. The voltage tap of the array is selected according to the value of bit 11, to which the result has been already set.

Bit 11 = 0: (1/4AV_{REFPn})
Bit 11 = 1: (3/4AV_{REFPn})

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 10 = 1
Analog input voltage ≤ Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR.

(12) When comparison of 12 bits has been completed, the valid digital value result remains in the successive approximation register (SAR). This value is transferred to A/Dn conversion result register m (ADnCRm) and the conversion result is stored in this register in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, and conversion channel specification mode (n = 0, 1, m = 0 to 15). The valid digital value is stored in the A/Dn conversion result extension buffer register a in the extension buffer mode, and is shifted to A/Dn conversion result extension register a when selection load trigger x is generated and stored (x = 1, 2, a = 0 to 4). When A/D conversion has ended the specified number of times, an A/Dn conversion end interrupt request signal (INTADn) is generated.

12.4.2 Input voltage and conversion result

The relationship between the analog voltage input to the analog input pin (ANInk) and the A/D conversion result (of A/Dn conversion result register m (ADnCRm) or A/Dn conversion result extension register a (ADnECRa)) is as follows:

$$\text{ADCR} = \text{INT}\left(\frac{V_{\text{IN}}}{AV_{\text{REFP}}} \times 4{,}096 + 0.5\right)$$

or,

$$(\text{ADCR} - 0.5) \times \frac{AV_{\text{REFP}}}{4{,}096} \leq V_{\text{IN}} < (\text{ADCR} + 0.5) \times \frac{AV_{\text{REFP}}}{4{,}096}$$

INT( ): Function that returns the integer of the value in ( )

V_{IN}: Analog input voltage

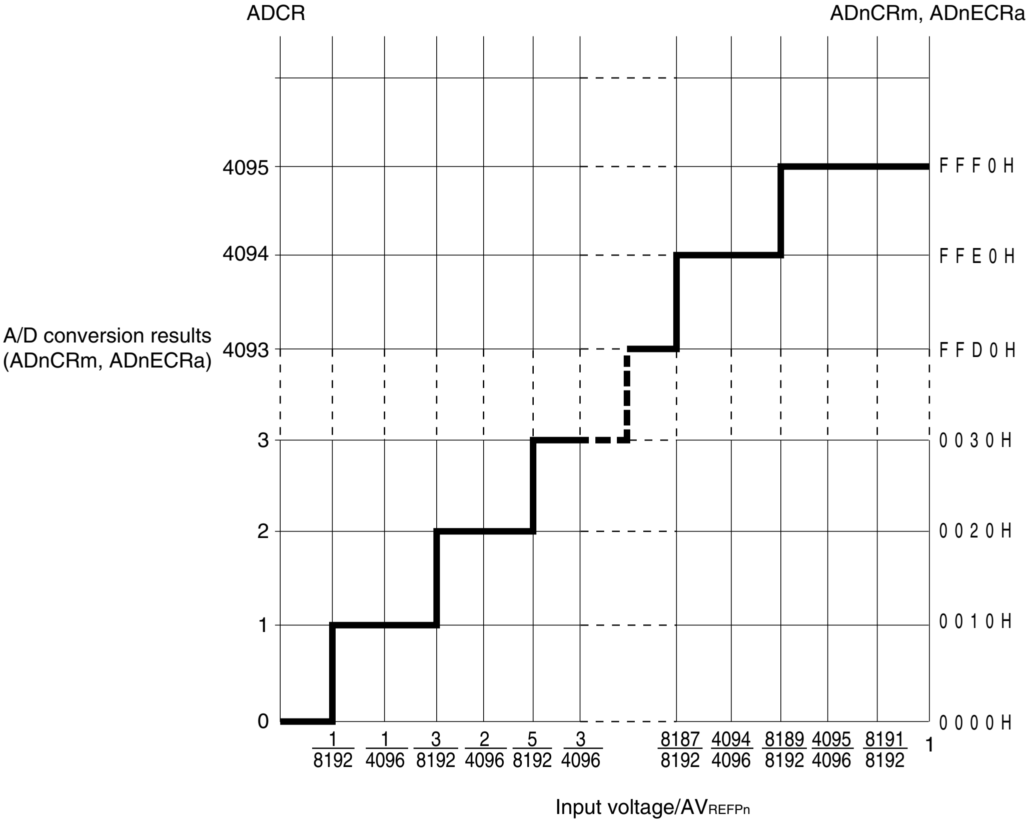
AV_{REFP}: AV_{REFPn} pin voltage

ADCR: Value of A/Dn conversion result register m (ADnCRm) or A/Dn conversion result extension register a (ADnECRa)

The relationship between the analog input voltage and the A/D conversion result is shown below.

Remark A/D converter 0: n = 0, m = 0 to 15, k = 0 to 3, 5 to 7, a = 0 to 4
A/D converter 1: n = 1, m = 0 to 15,
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7
a = 0 to 4

Figure 12-11. Relationship Between Analog Input Voltage and A/D Conversion Results



Remark A/D converter 0: n = 0, m = 0 to 15, k = 0 to 3, 5 to 7, a = 0 to 4

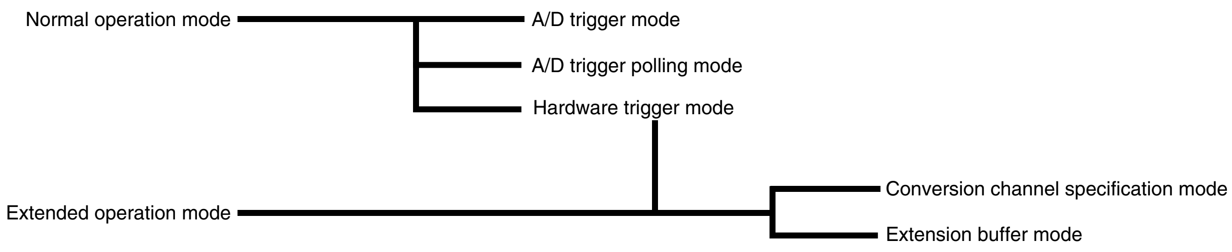
A/D converter 1: n = 1, m = 0 to 15,
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7
a = 0 to 4

12.4.3 Operation mode

Various conversion operations can be specified for the A/D converters 0 and 1 by specifying the operation mode. The operation mode is set by the ADnSCM, ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, ADLTS2, and ADnOCKS registers.

The following shows the relationship between the operation modes.

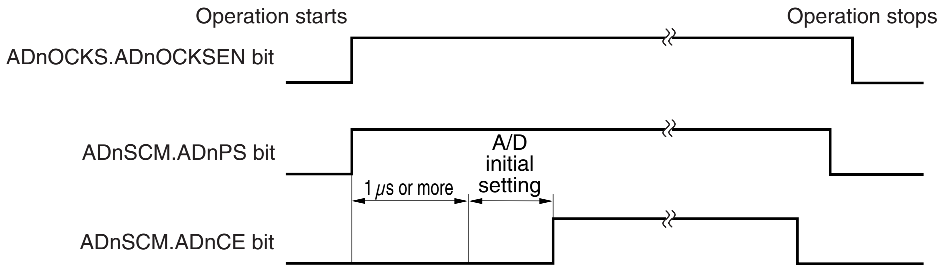
Remark n = 0, 1



Caution Be sure to set the hardware trigger mode when the conversion channel specification mode or extension buffer mode is used.

12.4.4 Operation setting

Start or stop the operation of A/D converters 0 and 1 in the following procedure.



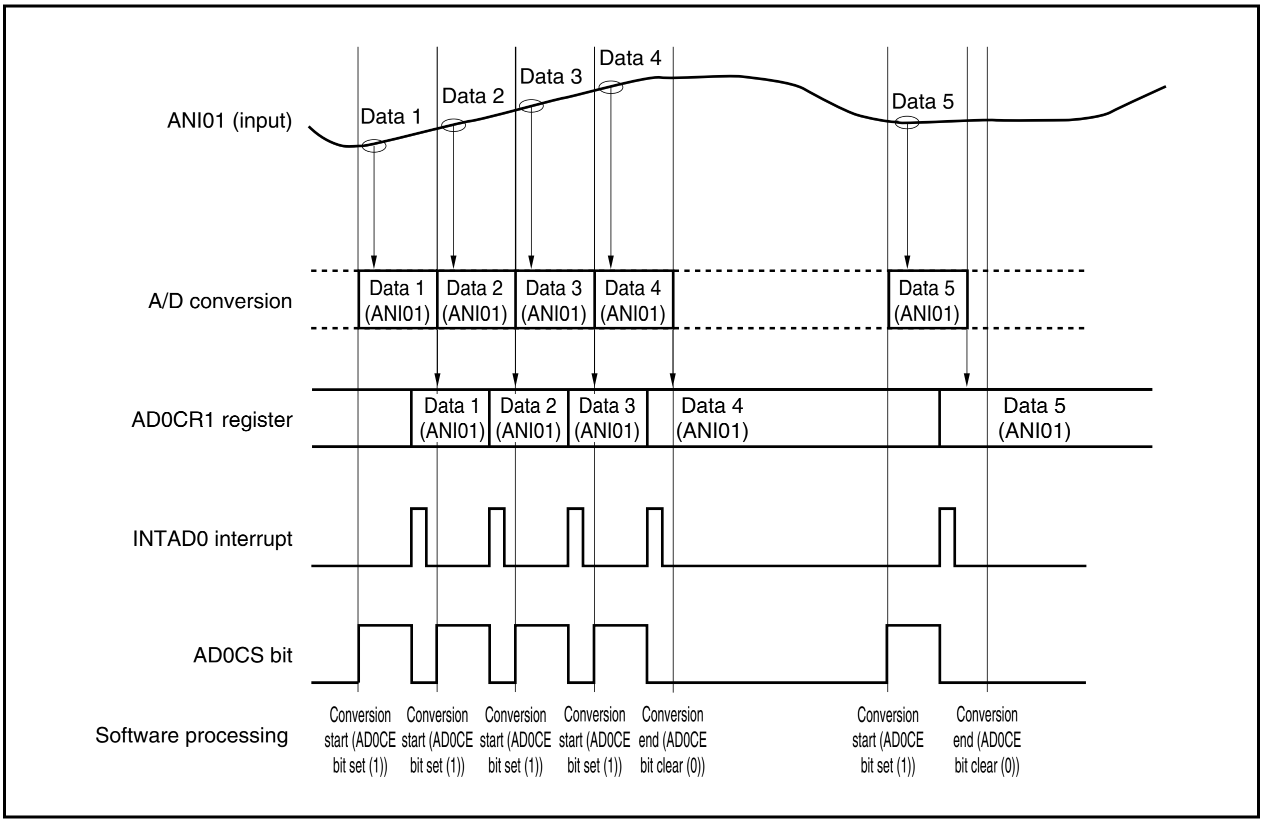
12.4.5 Operation of 1-channel conversion

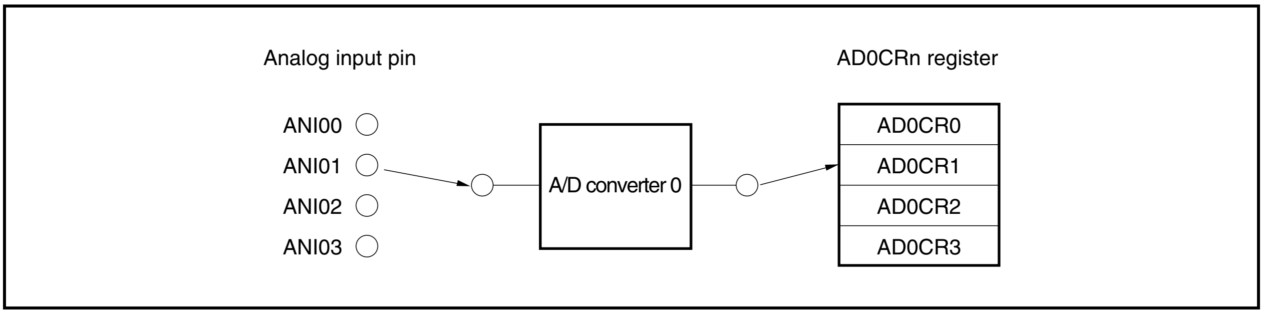
The signal of one analog input pin (ANInk) specified by the ADnCHEN register is converted. The result of conversion is stored in the ADnCRk register corresponding to the ANInk pin. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis, and an A/Dn conversion end interrupt request signal (INTADn) is generated each time conversion has been completed.

After completion of A/D conversion, the conversion operation is stopped in the A/D trigger mode or A/D trigger polling mode. In the hardware trigger mode, the A/D converter waits for a trigger.

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7

Figure 12-12. Operation of 1-Channel Conversion (in A/D Trigger Mode): A/D Converter 0





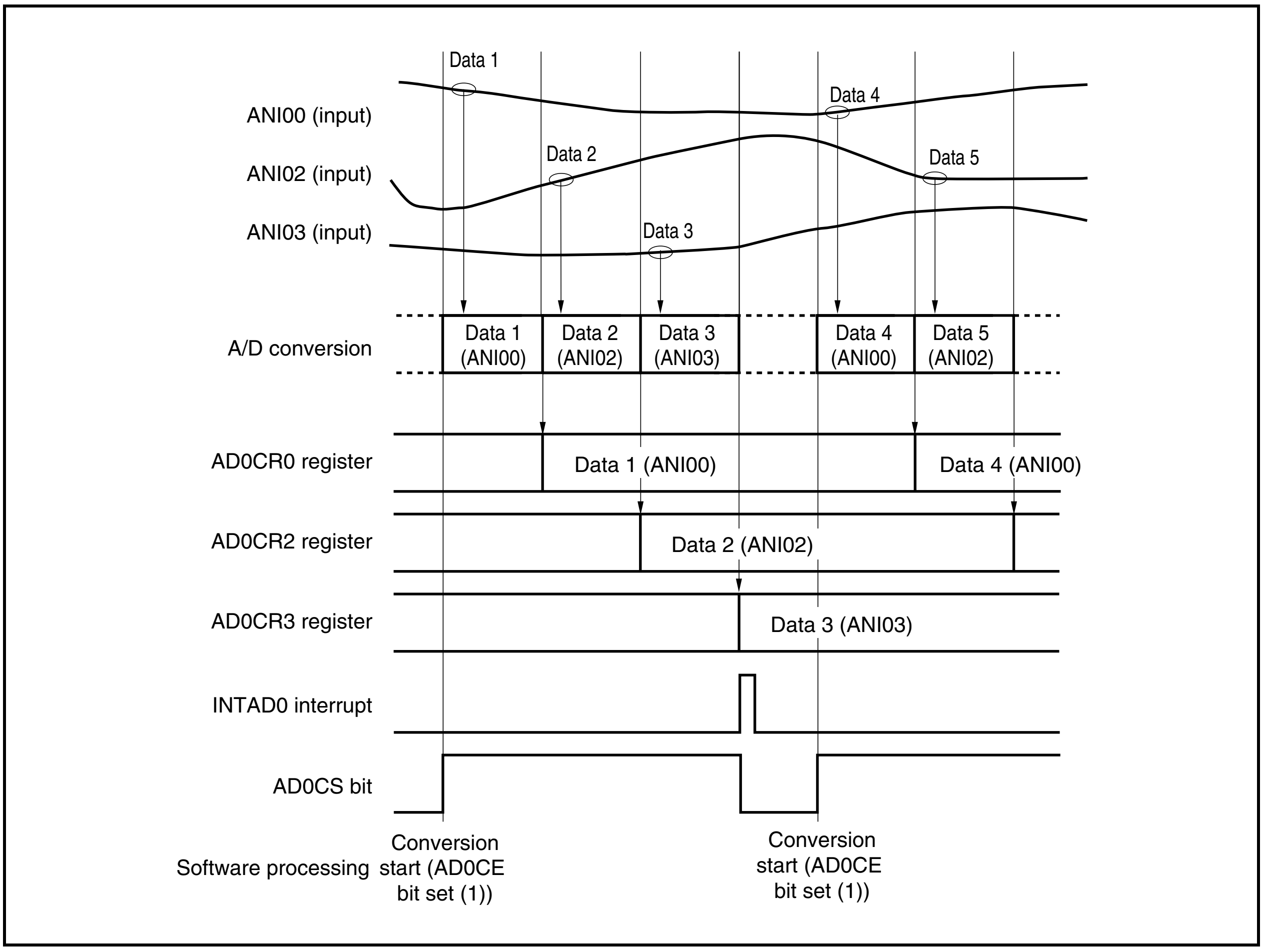
12.4.6 Operation of multiple channel conversion

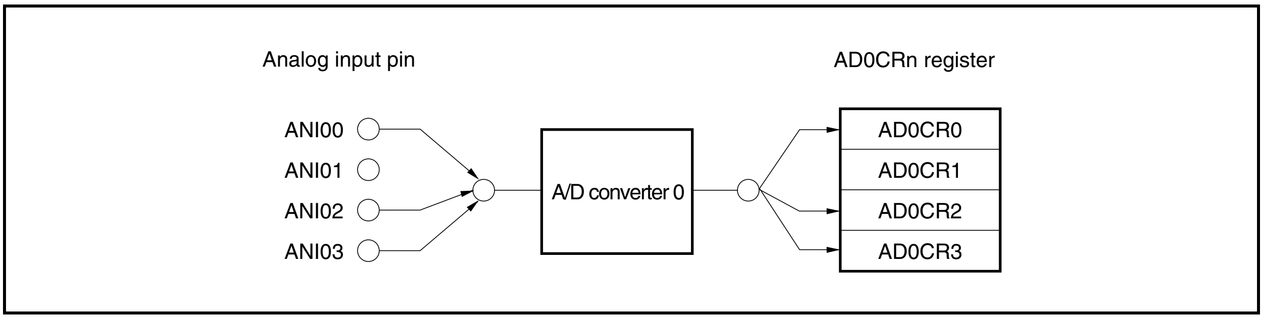
The signals of two or more analog input pins (ANInk) specified by the ADnCHEN register are converted. The signals are sequentially converted starting from the pin with the lowest number (in the example in Figure 12-13, ANI00 → ANI02 → ANI03). An analog input pin that is not specified is skipped. The result of conversion is stored in the ADnCRk register corresponding to the ANInk pin. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis. When conversion of the signal of the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated.

After completion of A/D conversion, the conversion operation is stopped in the A/D trigger mode or A/D trigger polling mode. In the hardware trigger mode, the A/D converter waits for a trigger.

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7

Figure 12-13. Operation of Multiple Channel Conversion (in A/D Trigger Mode): A/D Converter 0





12.4.7 A/D trigger mode (normal operation mode)

A/D conversion is started when the ADnSCM.ADnCE bit is set to 1.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion, the conversion is stopped and started again from the beginning.

(1) Operation of 1-channel conversion

The signal of one analog input pin (ANInk) is converted once and the result is stored in one ADnCRk register. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis.

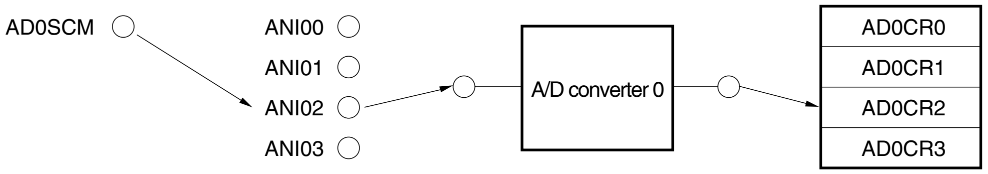
Each time conversion has been completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, The A/D converter stops conversion operation with the ADnSCM.ADnCE bit remaining set to 1. The A/D conversion can be restarted by setting the ADnCE bit to 1.

This operation is suitable for an application where the result of A/D conversion should be read each time conversion has been completed once.

Analog Input Pin	A/D Conversion Result Register
ANInk	ADnCRk

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7

Figure 12-14. Example of 1-Channel Conversion Operation (A/D Trigger Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Signal of ANI02 pin is A/D converted
(3) Conversion result is stored in AD0CR2 register
(4) AD0SCM.AD0CS bit = 0
(5) INTAD0 interrupt request signal is generated

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 000, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and AD0CHEN register = 0004H.

(2) Operation of multiple channel conversion

The signals of two or more analog input pins specified by the ADnCHEN register are converted sequentially starting from the pin with the lowest number. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, the A/D converter stops conversion operation with the ADnSCM.ADnCE bit remaining set to 1. The A/D conversion can be restarted by setting the ADnCE bit to 1.

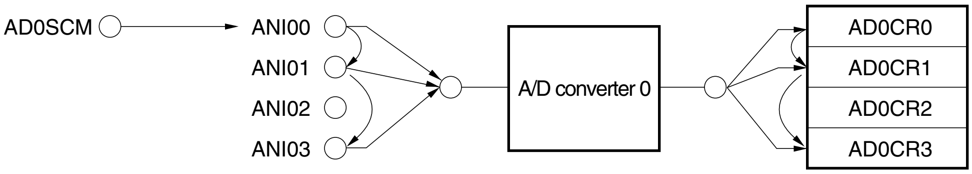
This operation is suitable for an application where two or more analog input signals should be monitored.

Analog Input Pin	A/D Conversion Result Register
ANInk[Note]	ADnCRk
⋮	⋮
ANInk[Note]	ADnCRk

Note Two or more can be specified by the ADnCHEN register.
However, A/D conversion is sequentially executed starting from the pin with the lowest number.

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7

Figure 12-15. Example of Multiple Channel Conversion Operation (A/D Trigger Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Signal of ANI00 pin is A/D-converted
(3) Conversion result is stored in AD0CR0 register
(4) Signal of ANI01 pin is A/D-converted
(5) Conversion result is stored in AD0CR1 register
(6) Signal of ANI03 pin is A/D-converted
(7) Conversion result is stored in AD0CR3 register
(8) AD0SCM.AD0CS bit = 0
(9) INTAD0 interrupt request signal is generated

Remark This is an operation example when the
AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 000,
AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and
AD0CHEN register = 000BH.

12.4.8 A/D trigger polling mode (normal operation mode)

A/D conversion is started when the ADnSCM.ADnCE bit is set to 1.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

In the A/D trigger polling mode, it is not necessary to write 1 to the ADnCE bit to restart A/D conversion after the A/Dn conversion end interrupt request signal (INTADn) is generated.

If the ADnSCM register is written during A/D conversion, the conversion is stopped and started again from the beginning.

(1) Operation of 1-channel conversion

The signal of one analog input pin (ANInk) is converted once and the result is stored one ADnCRk register. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis.

Each time conversion has been completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. A/D conversion is repeated until the ADnSCM.ADnCE bit is set to 0. The conversion operation is stopped when the ADnCE bit is cleared to 0.

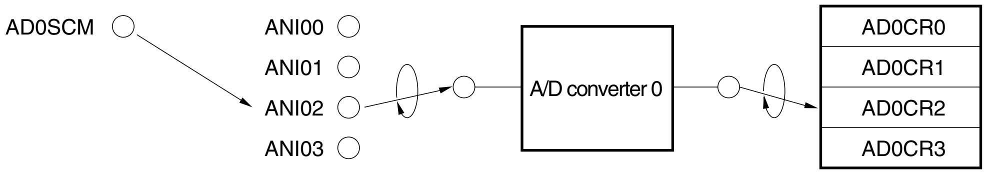
It is not necessary to set the ADnCE bit to restart the conversion operation in the A/D trigger polling mode.

This operation is suitable for an application where the A/D conversion value is always read.

Analog Input Pin	A/D Conversion Result Register
ANInk	ADnCRk

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7

Figure 12-16. Example of 1-Channel Conversion Operation (A/D Trigger Polling Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Signal of ANI02 pin is A/D-converted
(3) Conversion result is stored in AD0CR2 register
(4) AD0SCM.AD0CS bit = 0
(5) INTAD0 interrupt request signal is generated
(6) Return to (2)
(7) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the
AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 100,
AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and
AD0CHEN register = 0004H.

(2) Operation of multiple channel conversion

The signals of two or more analog input pins specified by the ADnCHEN register are converted sequentially starting from the pin with the lowest number. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. A/D conversion is repeated until the ADnSCM.ADnCE bit is set to 0. The conversion operation is stopped when the ADnCE bit is cleared to 0.

It is not necessary to set the ADnCE bit to restart the conversion operation in the A/D trigger polling mode.

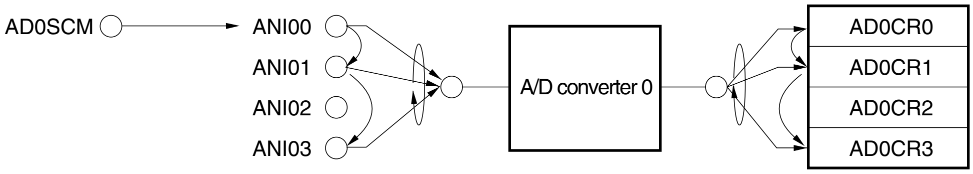
This operation is suitable for an application where the A/D conversion value is always read.

Analog Input Pin	A/D Conversion Result Register
ANInk[Note]	ADnCRk
⋮	⋮
ANInk[Note]	ADnCRk

Note Two or more can be specified by the ADnCHEN register.
However, A/D conversion is sequentially executed starting from the pin with the lowest number.

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7

Figure 12-17. Example of Multiple Channel Conversion Operation (A/D Trigger Polling Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Signal of ANI00 pin is A/D-converted
(3) Conversion result is stored in AD0CR0 register
(4) Signal of ANI01 pin is A/D-converted
(5) Conversion result is stored in AD0CR1 register
(6) Signal of ANI03 pin is A/D-converted
(7) Conversion result is stored in AD0CR3 register
(8) AD0SCM.AD0CS bit = 0
(9) INTAD0 interrupt request signal is generated
(10) Return to (2)
(11) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the
AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 100,
AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and
AD0CHEN register = 000BH.

12.4.9 Hardware trigger mode (normal operation mode)

The A/D converter waits for a trigger when the ADnSCM.ADnCE bit is set to 1, and starts A/D conversion when a trigger specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits is generated.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion, the conversion is stopped and becomes trigger wait status again.

(1) Operation of 1-channel conversion

The signal of one analog input pin (ANInk) is converted once, using a signal specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits as a trigger, and the result of conversion is stored in one ADnCRk register. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis.

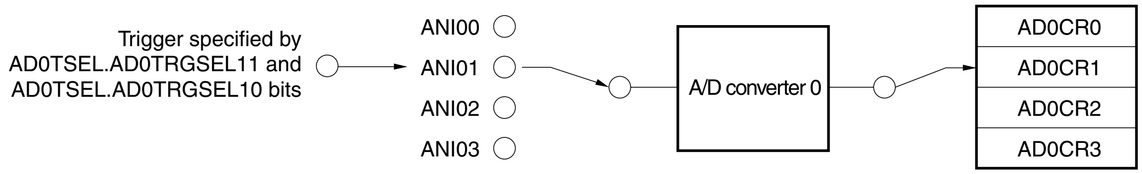
Each time conversion has been completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After completing the conversion, the converter waits for the trigger with the ADnSCM.ADnCE bit set to 1.

This operation is suitable for an application where the result of A/D conversion should be read each time conversion by one trigger has been completed.

Analog Input Pin	A/D Conversion Result Register
ANInk	ADnCRk

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7

Figure 12-18. Example of 1-Channel Conversion Operation (Hardware Trigger Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Trigger specified by AD0TSEL.AD0TRGSEL11 and AD0TSEL.AD0TRGSEL10 bits is generated
(3) Signal of ANI01 pin is A/D-converted
(4) Conversion result is stored in AD0CR1 register
(5) AD0SCM.AD0CS bit = 0
(6) INTAD0 interrupt request signal is generated
(7) Returns to (3) when the next trigger is input
(8) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 001, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and AD0CHEN register = 0002H.

(2) Operation of multiple channel conversion

The signals of two or more analog input pins specified by the ADnCHEN register are sequentially converted, starting from the pin with the lowest number, using a signal specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits as a trigger. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After completion of conversion, the A/D converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

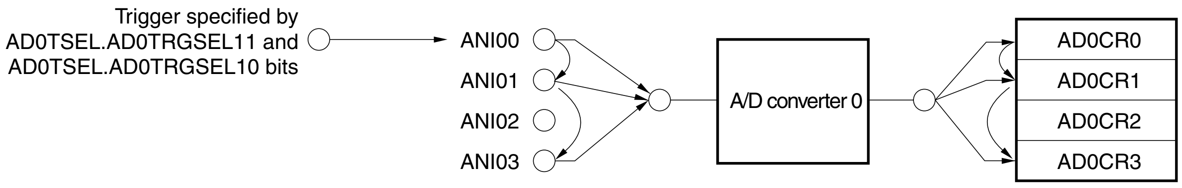
This operation is suitable for an application where two or more analog input signals should be monitored when the trigger is generated.

Analog Input Pin	A/D Conversion Result Register
ANInk Note	ADnCRk
⋮	⋮
ANInk Note	ADnCRk

Note Two or more can be specified by the ADnCHEN register.
However, A/D conversion is sequentially executed starting from the pin with the lowest number.

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7

Figure 12-19. Example of Multiple Channel Conversion Operation (Hardware Trigger Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Trigger specified by AD0TSEL.AD0TRGSEL11 and AD0TSEL.AD0TRGSEL10 bits is generated
(3) Signal of ANI00 pin is A/D-converted
(4) Conversion result is stored in AD0CR0 register
(5) Signal of ANI01 pin is A/D-converted
(6) Conversion result is stored in AD0CR1 register
(7) Signal of ANI03 pin is A/D-converted
(8) Conversion result is stored in AD0CR3 register
(9) AD0SCM.AD0CS bit = 0
(10) INTAD0 interrupt request signal is generated
(11) Returns to (3) when the next trigger is input
(12) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the
AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 001,
AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and
AD0CHEN register = 000BH.

12.4.10 Conversion channel specification mode (extension operation mode)

When the ADnSCM.ADnCE bit is set to 1, the A/D converter waits for a trigger. When selection trigger 1 specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits is generated, the converter starts A/D conversion.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion operation, the conversion is stopped and the converter waits for the trigger again.

The analog input pin is specified by the ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 and ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits. Each time selection trigger 1 is generated, the analog input pins specified by the ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 and ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits are sequentially selected.

The signal of a specified analog input pin is converted the number of times specified by the ADnCHEN register (up to 16 times), using selection trigger 1 as the trigger, and the result is stored in the ADnCRm register specified by the ADnCHEN register. The conversion results are sequentially stored from ADnCR0.

When the signal of the specified analog input pin has been converted the number of times (up to 16 times) specified by the ADnCHEN register, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, the A/D converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

This operation is suitable for an application where two or more analog input signals should be monitored.

Selection Trigger	Analog Input Pin	A/D Conversion Result Extension Register
Selection trigger 1	ANInx[Note 1]	ADnCR0[Note 3]
	ANInx[Note 1]	|
	ANInx[Note 1]	ADnCRm[Note 3]
Selection trigger 2	ANIny[Note 2]	ADnCR0[Note 3]
	ANIny[Note 2]	|
	ANIny[Note 2]	ADnCRm[Note 3]

Notes 1. Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits

2. Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits

3. Two or more times can be set by the ADnCHEN register.

Cautions 1. Be sure to set the hardware trigger mode as the conversion channel specification mode.

2. Be sure to set the ADnCHEN register using the lower bits, justifying to the bottom. Any other setting is prohibited.

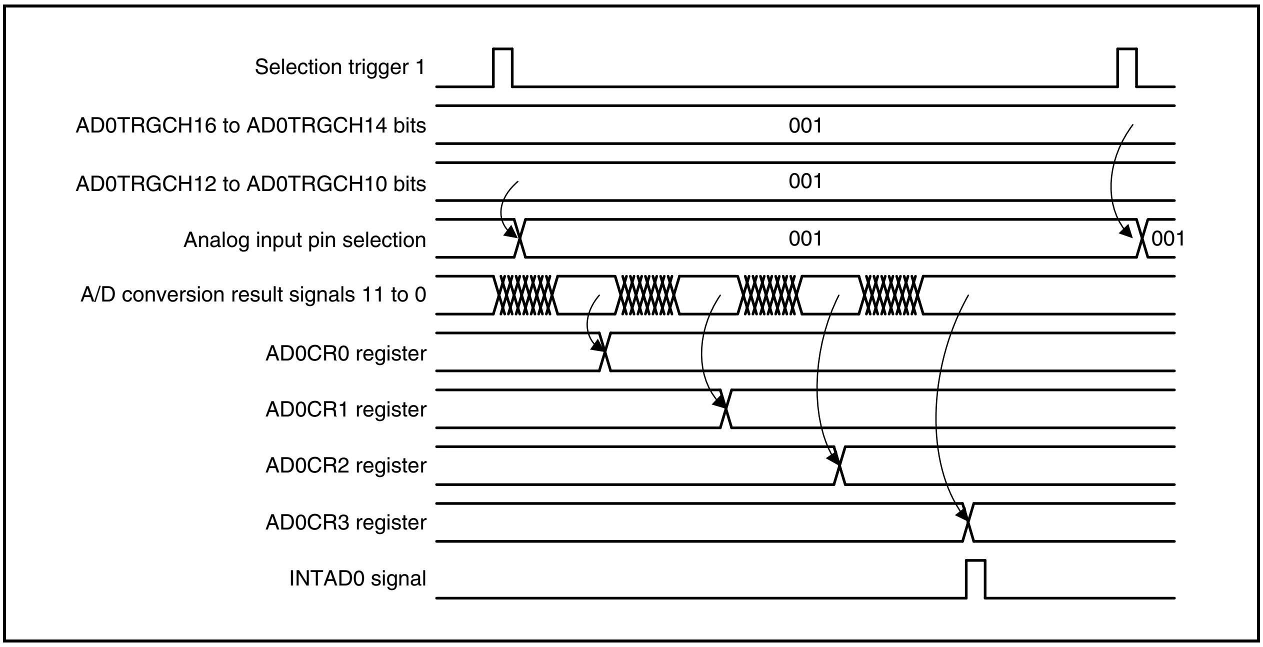
3. Setting of the ADnCH2 register is invalid.

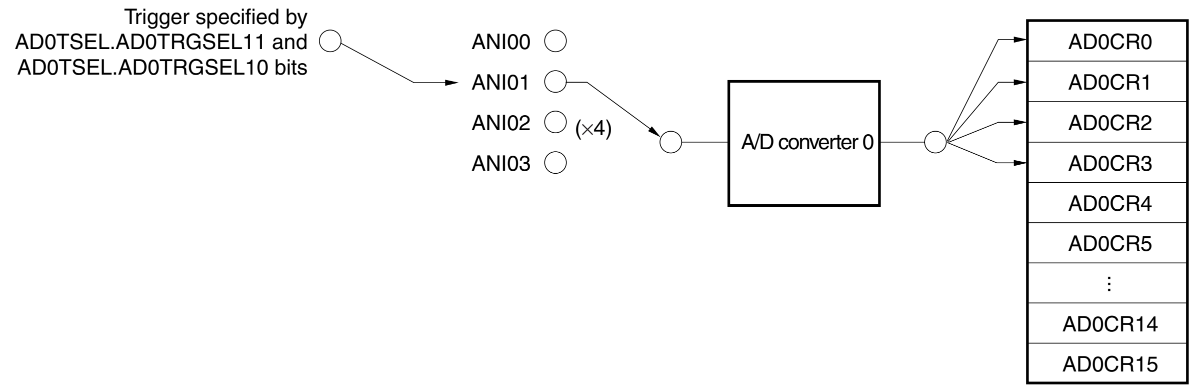
4. The ADnECRa, ADnECRaH, ADnFLG, and ADnFLGB registers are not used. If these registers are read, 0000H and 00H are read.

5. Selection trigger 1 is ignored if it is generated during A/D conversion operation. The next selection trigger 1 is accepted when a trigger is generated after completion of A/D conversion (after generation of the INTADn signal).

Remark A/D converter 0: n = 0
k = 0 to 3, 5 to 7
m = 0 to 15
A/D converter 1: n = 1
V850E/IG4-H: k = 0 to 2, 5 to 7
V850E/IH4-H: k = 0 to 3, 5 to 7
m = 0 to 15

Figure 12-20. Example of Operation in Conversion Channel Specification Mode: A/D Converter 0


Selection trigger 1
AD0TRGCH16 to AD0TRGCH14 bits
001
AD0TRGCH12 to AD0TRGCH10 bits
001
Analog input pin selection
001
001
A/D conversion result signals 11 to 0
AD0CR0 register
AD0CR1 register
AD0CR2 register
AD0CR3 register
INTAD0 signal



(1) AD0CE bit = 1 (enable)

(2) Trigger specified by AD0TSEL.AD0TRGSEL11 and AD0TSEL.AD0TRGSEL10 bits is generated

(3) Signal of ANI01 pin is A/D-converted

(4) Conversion result is stored in AD0CR0 register

(5) Signal of ANI01 pin is A/D-converted

(6) Conversion result is stored in AD0CR1 register

(7) Signal of ANI01 pin is A/D-converted

(8) Conversion result is stored in AD0CR2 register

(9) Signal of ANI01 pin is A/D-converted

(10) Conversion result is stored in AD0CR3 register

(11) AD0SCM.AD0CS bit = 0

(12) INTAD0 interrupt request signal is generated

(13) Returns to (3) when the next trigger is input

(14) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the
AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 001,
AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 10,
AD0CHEN register = 000FH,
AD0CH1.AD0TRGCH12 to AD0CH1.AD0TRGCH10 bits = 001, and
AD0CH1.AD0TRGCH16 to AD0CH1.AD0TRGCH14 bits = 001.

12.4.11 Extension buffer mode (extension operation mode)

When the ADnSCM.ADnCE bit is set to 1, the A/D converter waits for a trigger. When selection trigger 1 specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits or selection trigger 2 specified by the ADnTSEL.ADnTRGSEL21 and ADnTSEL.ADnTRGSEL20 bits is generated, the converter starts A/D conversion.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion operation, the conversion is stopped and the converter waits for the trigger again.

The analog input pin for selection trigger x is specified by the ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 and ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits. Each time selection trigger x is generated, the analog input pins specified by the ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 and ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits are sequentially selected.

When selection trigger 1 is used, the signal of the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits is converted when the trigger is generated for the first time. The result is stored in the A/Dn conversion result extension buffer register 0 and an A/Dn conversion end interrupt request signal (INTADn) is generated. When the trigger is generated the second time, the signal of the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits is converted. The result is stored in the A/Dn conversion result extension buffer register 0 and, at the same time, the first value stored in the A/Dn conversion result extension buffer register 0 is stored in the A/Dn conversion result extension buffer register 1. Then the INTADn interrupt request signal is generated. For A/D conversion using selection trigger 1, up to three A/Dn conversion result extension buffer registers, 0 to 2, can be used. When selection load trigger 1 is later generated, the values of the A/Dn conversion result extension buffer registers 0 to 2 are transferred to the ADnECR0 to ADnECR2 registers. After A/D conversion is competed, the converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

When selection trigger 2 is used, the signal of the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits is converted when the trigger is generated for the first time, and the result is stored in the A/Dn conversion end extension buffer register 3. Then an A/Dn conversion end interrupt request signal (INTADn) is generated. When the trigger is generated the second time, the signal of the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits is converted and the result is stored in the A/Dn conversion result extension buffer register 4. At the same time, the value stored first in the A/Dn conversion result extension buffer register 3 is stored in the A/Dn conversion result extension buffer register 4, and the INTADn interrupt request signal is generated. When selection trigger 2 is used for A/D conversion, up to two A/Dn conversion result extension buffer registers, 3 and 4, can be used. When selection load trigger 2 is generated again, the values of the A/Dn conversion result extension buffer registers 3 and 4 are transferred to and stored in the ADnECR3 and ADnECR4 registers. After A/D conversion is completed, the converter waits for the trigger with the ADnCE bit remaining set to 1.

Therefore, the contents of the ADnECR0 to ADnECR4 registers can be saved to RAM all at once.

This operation is suitable for an application where there is little time to save the conversion result and two or more analog input signals should be monitored when a trigger is generated.

Selection Trigger	Analog Input Pin	A/D Conversion Result Extension Register
Selection trigger 1	ANInx[Note 1]	ADnECR0 to ADnECR2
Selection trigger 1	ANIny[Note 2]	ADnECR0, ADnECR1
Selection trigger 1	ANInx[Note 1]	ADnECR0
Selection trigger 2	ANIns[Note 3]	ADnECR3, ADnECR4
Selection trigger 2	ANInt[Note 4]	ADnECR3

Notes **1.** Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits

2. Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits

3. Set by ADnCH2.ADnTRGCH22 to ADnCH2.ADnTRGCH20 bits

4. Set by ADnCH2.ADnTRGCH26 to ADnCH2.ADnTRGCH24 bits

Cautions 1. In the extension buffer mode, be sure to set the hardware trigger mode and the ADnCHEN register to 0001H.

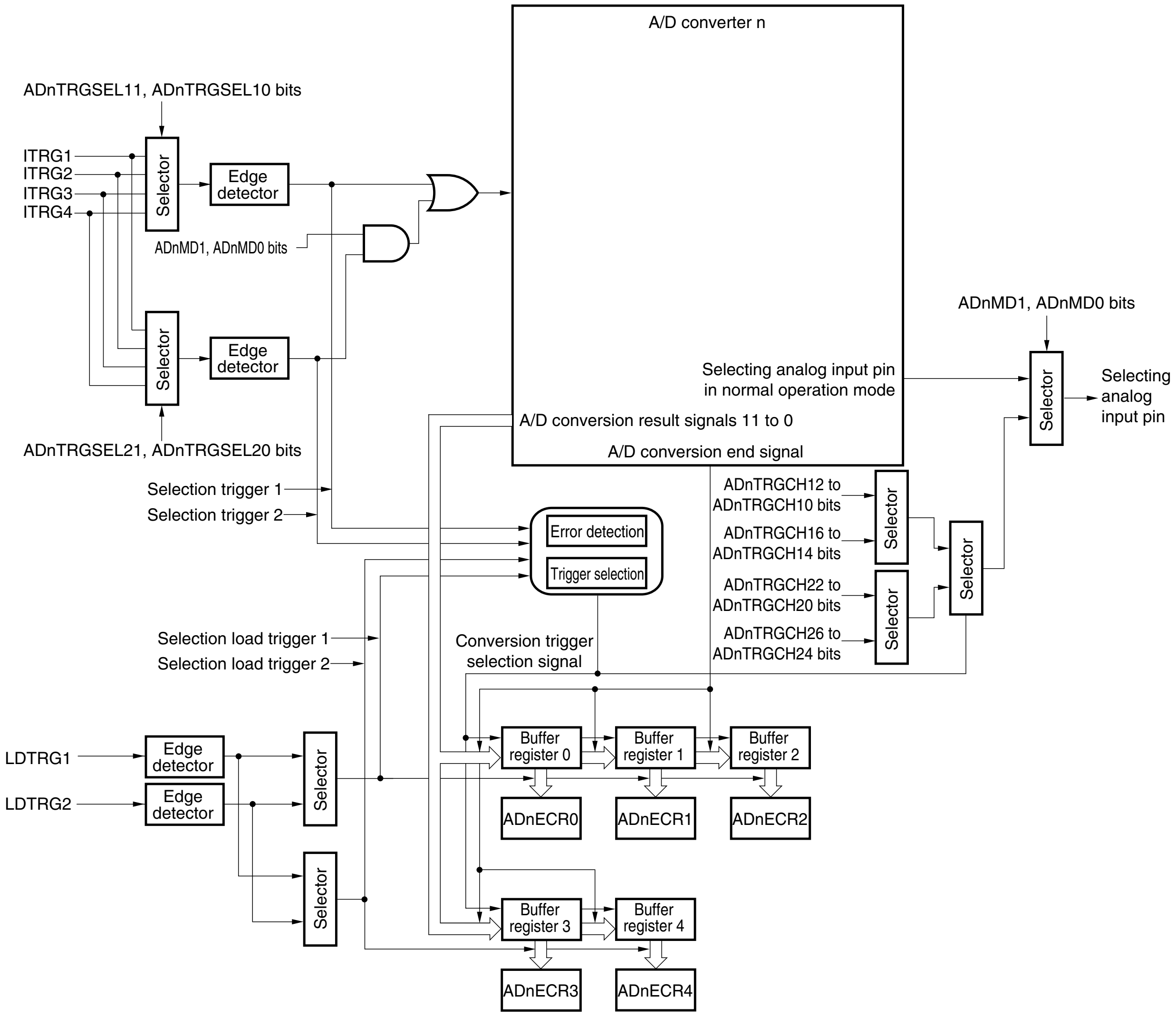
2. The conversion result is stored in the ADnECRa register. The value of the ADnCRm register is undefined.

Remark n = 0, 1

a = 0 to 4

m = 1, 2

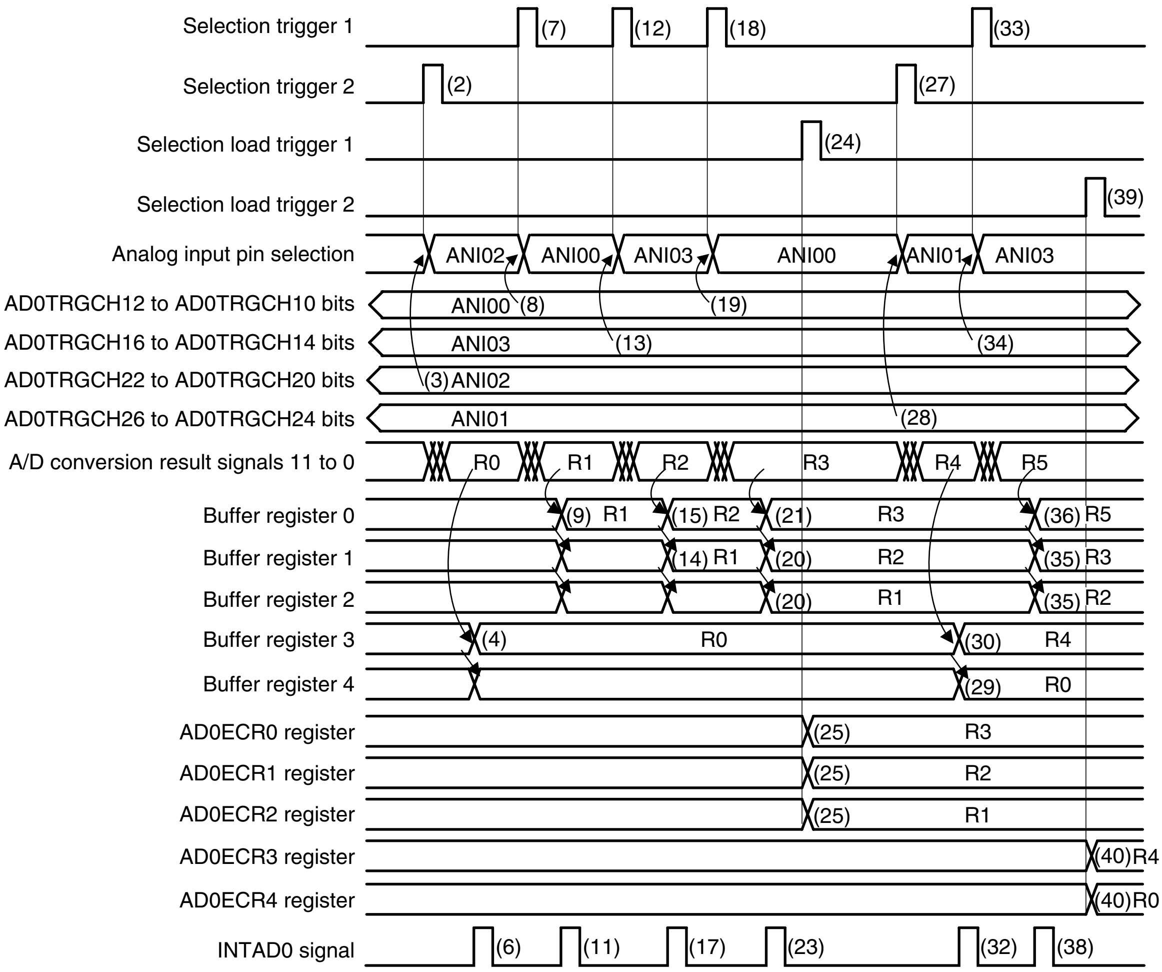
Figure 12-21. Block Diagram in Extension Buffer Mode



Remarks 1. Buffer registers 0 to 4: A/Dn conversion result extension buffer registers 0 to 4

2. n = 0, 1

Figure 12-22. Example of Operation in Extension Buffer Mode: A/D Converter 0 (1/2)



Remarks 1. Buffer registers 0 to 4: A/Dn conversion result extension buffer registers 0 to 4

2. R0 to R6: Conversion result

3. n = 0, 1

Figure 12-22. Example of Operation in Extension Buffer Mode: A/D Converter 0 (2/2)

(1) AD0CE bit = 1 (enable)
(2) Selection trigger 2 is generated
(3) Signal of ANI02 pin is A/D-converted
(4) Conversion result is stored in buffer register 3
(5) AD0SCM.AD0CS bit = 0
(6) INTAD0 interrupt request signal is generated
(7) Selection trigger 1 is generated
(8) Signal of ANI00 pin is A/D-converted
(9) Conversion result is stored in buffer register 0
(10) AD0SCM.AD0CS bit = 0
(11) INTAD0 interrupt request signal is generated
(12) Selection trigger 1 is generated
(13) Signal of ANI03 pin is A/D-converted
(14) Shifted from buffer register 0 to buffer register 1
(15) Conversion result is stored in buffer register 0
(16) AD0SCM.AD0CS bit = 0
(17) INTAD0 interrupt request signal is generated
(18) Selection trigger 1 is generated
(19) Signal of ANI00 pin is A/D-converted
(20) Shifted from buffer register 0 to buffer register 1 to buffer register 2
(21) Conversion result is stored in buffer register 0
(22) AD0SCM.AD0CS bit = 0
(23) INTAD0 interrupt request signal is generated
(24) Selection load trigger 1 is generated
(25) Shifted from buffer registers 0 to 2, to AD0ECR0 to AD0ECR2
(26) AD0SCM.AD0CS bit = 0
(27) Selection trigger 2 is generated
(28) Signal of ANI01 pin is A/D-converted
(29) Shifted from buffer register 3 to buffer register 4
(30) Conversion result is stored in buffer register 3
(31) AD0SCM.AD0CS bit = 0
(32) INTAD0 interrupt request signal is generated
(33) Selection trigger 1 is generated
(34) Signal of ANI03 pin is A/D-converted
(35) Shifted from buffer register 0 to buffer register 1 to buffer register 2
(36) Conversion result is stored in buffer register 0
(37) AD0SCM.AD0CS bit = 0
(38) INTAD0 interrupt request signal is generated
(39) Selection load trigger 2 is generated
(40) Shifted from buffer registers 3 and 4 to AD0ECR3 and AD0ECR4 registers
(41) AD0SCM.AD0CS bit = 0
(42) When the next trigger is input, the operation is performed in accordance with that trigger.
(43) Set ADnCE bit to 0 to end (stop)

(1) Error detection function

The extension buffer mode has an error detection function. If a trigger (selection trigger 1, selection trigger 2, selection load trigger 1, or selection load trigger 2) is generated during A/D conversion, an error occurs. The error is detected by the ADnFLG.ADnTERR2 and ADnFLG.ADnTERR1 flags, and ADnFLGB.ADnTERRB2 and ADnFLGB.ADnTERRB1 flags.

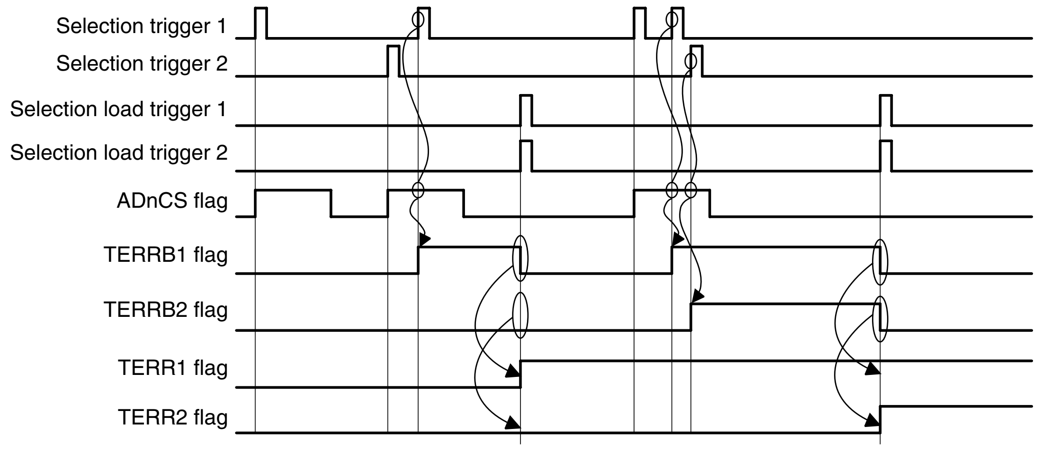
Cautions 1. Selection trigger 1, selection trigger 2, selection load trigger 1, and selection load trigger 2 are generated when asynchronous signals ITRG1 to ITRG4, LDTRG1, and LDTRG2 signals are synchronized. Although the timing of inputting these triggers seems to be the same, their simultaneous operation is not guaranteed because the asynchronous signals are synchronized.

2. Selection trigger 1 or 2 is ignored, even if it is generated again, during a period of up to 2.5 base clocks (f_{AD01}) after the trigger is once generated (no error occurs).

(a) Error detection by generation of selection trigger 1 or 2 during A/D conversion

If selection trigger 1 is generated during A/D conversion, the ADnFLGB.ADnTERRB1 flag is set to 1 and A/D conversion by selection trigger 1 is ignored. If selection load trigger 1 is generated next, the value of the ADnTERRB1 flag is stored in the ADnFLG.ADnTERR1 flag.

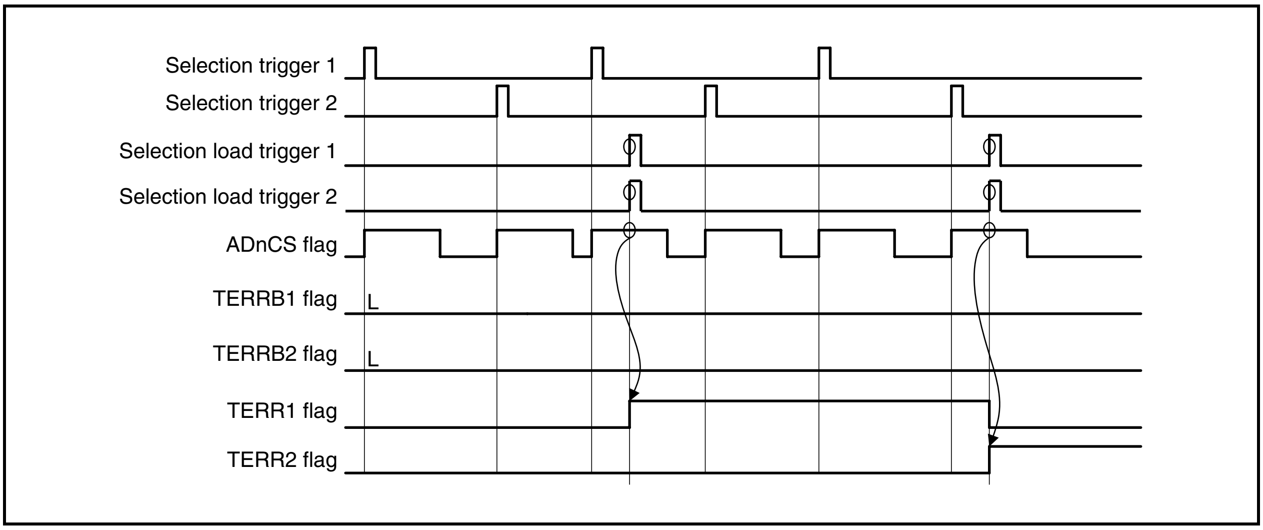
Similarly, if selection trigger 2 is generated during A/D conversion, the ADnFLGB.ADnTERRB2 flag is set to 1 and A/D conversion by selection trigger 2 is ignored. When selection load trigger 2 is generated next, the value of the ADnTERRB2 flag is stored in the ADnFLG.ADnTERR2 flag.



(b) Error detection by generation of selection load trigger 1 or 2 during A/D conversion

If selection load trigger 1 is generated during A/D conversion that uses selection trigger 1, the ADnFLG.ADnTERR1 flag is set to 1. A/D conversion and load operation are performed normally.

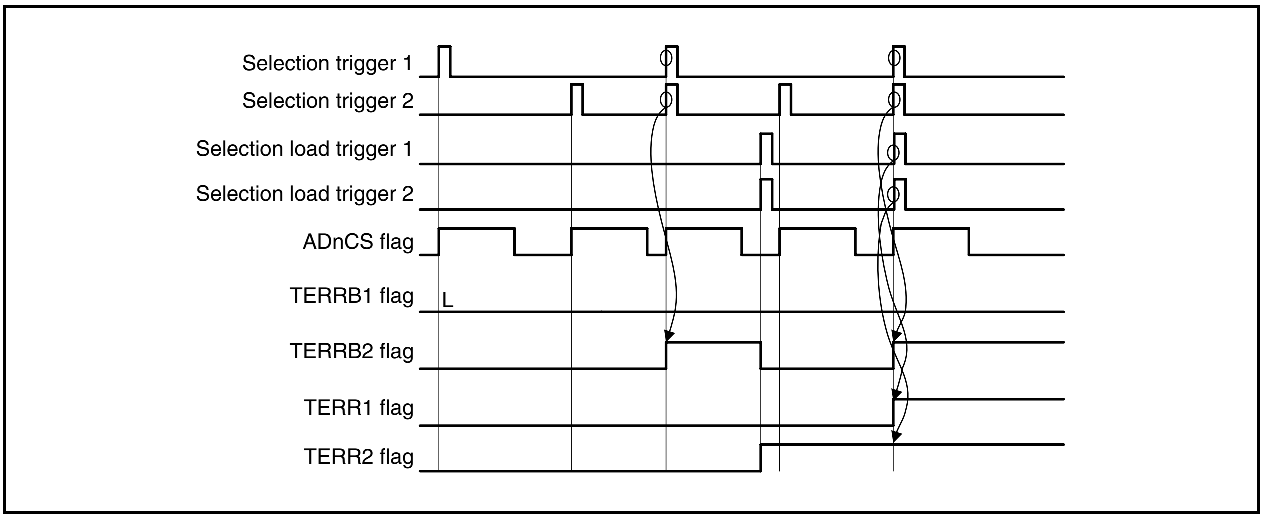
Similarly, if selection load trigger 2 is generated during A/D conversion that uses selection trigger 2, the ADnTERR2 flag is set to 1. A/D conversion and load operation are performed normally.



(c) Error detection by simultaneous generation of selection triggers 1 and 2, and of selection triggers 1 and 2, and selection load triggers 1 and 2

If selection triggers 1 and 2 are simultaneously generated, A/D conversion that uses selection trigger 1 is started and selection trigger 2 is ignored. Therefore, the ADnFLGB.ADnTERRB2 flag is set to 1.

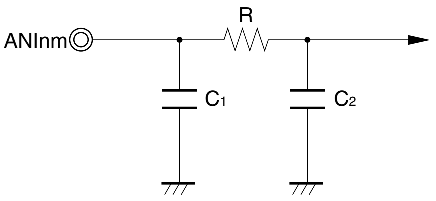
If selection triggers 1 and 2, and selection load triggers 1 and 2 are simultaneously generated, the ADnFLGB.ADnTERRB2, ADnFLG.ADnTERR1, and ADnFLG.ADnTERR2 flags are set to 1. A/D conversion by selection trigger 1 and load operation of selection load triggers 1 and 2 are performed normally. Selection trigger 2 is ignored.



<R>

12.5 Internal Equivalent Circuit

The following figure shows the equivalent circuit of the analog input block.



R	C_1	C_2
5.1 kΩ	15 pF	3.9 pF

Remarks 1. The maximum values are shown (reference values).

2. V850E/IG4-H: n = 0, 1
m = 0 to 3, 5 to 7 when n = 0
m = 0 to 2, 5 to 7 when n = 1
V850E/IH4-H: n = 0, 1
m = 0 to 3, 5 to 7

ADnCTC register				Number of A/D conversion clocks (f_{AD01})	Number of Sampling clocks (f_{AD01})
ADnFR3 bit	ADnFR2 bit	ADnFR1 bit	ADnFR0 bit		
0	0	0	0	89	69.5
0	0	0	1	88	68.5
0	0	1	0	57	37.5
0	0	1	1	56	36.5
0	1	0	0	41	21.5
0	1	0	1	40	20.5
0	1	1	0	35	15.5
0	1	1	1	34	14.5
1	0	0	0	34	14.5
1	0	0	1	33	13.5
1	0	1	0	33	13.5
1	0	1	1	32	12.5
1	1	0	0	32	12.5
1	1	0	1	31	11.5
1	1	1	0	31	11.5
1	1	1	1	30	10.5

12.6 Cautions

12.6.1 Stopping conversion operation

The ongoing conversion operation is stopped when 0 is written to the ADnSCM.ADnCE bit. At this time, the conversion result in the A/Dn conversion result register m (ADnCRm) and A/Dn conversion result extension register a (ADnECRa) is undefined. Therefore, read the A/D conversion result after A/D conversion has been completed (after the A/Dn conversion end interrupt request signal (INTADn) has been issued), and then write 0 to the ADnCE bit as necessary.

Note that the ADnCE bit is not cleared to 0 in all the modes even after the INTADn signal is generated.

Remark n = 0, 1
m = 0 to 15

12.6.2 Interval of trigger during conversion operation in hardware trigger mode, conversion channel specification mode, and extension buffer mode

Inputting a trigger during conversion operation is ignored in the hardware trigger mode, conversion channel specification mode, and extension buffer mode. Therefore, the interval of the trigger (input time) in the hardware trigger mode, conversion channel specification mode, and extension buffer mode must be longer than the A/D conversion time specified by the ADnCTC.ADnFR3 to ADnCTC.ADnFR0 bits (see **Table 12-2 Number of A/D Conversion Clocks and A/D Conversion Time**).

Remark n = 0, 1

12.6.3 Writing to ADnSCM register

(1) Restarting A/D conversion

To restart A/D conversion, write the same value to the ADnSCM register. To change the ADnPLM, ADnTRG1, and ADnTRG0 bits, be sure to set the ADnCE bit to 0.

(2) Contention between end of A/D conversion and writing to ADnSCM register

If completion of A/D conversion contends with writing to the ADnSCM register during A/D conversion operation, the conversion result is correctly stored in the ADnCRm and ADnECRa registers, if the A/Dn conversion end interrupt request signal (INTADn) is generated. If the INTADn signal is not generated, the A/D conversion operation is aborted. Therefore, the previous conversion result is held by the ADnCRm and ADnECRa registers.

(3) Successive writing to ADnSCM register

To successively write the ADnSCM register when the conversion operation is enabled (ADnCE bit = 1), be sure to wait for time of at least 5 base clocks (f_{AD01}).

The ADnSCM register can be successively written when the ADnCE bit is set to 1 after the ADnSCM register is written while the ADnCE bit = 0.

Remark n = 0, 1

12.6.4 A/D conversion start timing

In the conversion channel specification mode and extension buffer mode, starting A/D conversion is delayed up to 1.5 base clocks (f_{AD01}) as compared with the normal operation mode.

12.6.5 Operation in standby mode

(1) HALT mode

The A/D conversion operation continues. If the HALT mode is released by a maskable interrupt request signal that is not masked, the values of the ADnSCM, ADnCRm, and ADnECRa registers are held.

(2) IDLE mode and STOP mode

No conversion operation is performed because clock supply to A/D converters 0 and 1 is stopped.

Be sure to set the ADnSCM.ADnCE bit to 0 when the IDLE or STOP mode is set. At this time, setting the A/D power save mode (ADnSCM.ADnPS bit = 0) is recommended.

Remark n = 0, 1
m = 0 to 15

12.6.6 Timing of accepting trigger in conversion channel specification mode and extension buffer mode

In the conversion channel specification mode and extension buffer mode, selection trigger 1 or 2 is ignored, even if it is generated again, until the A/Dn conversion end interrupt signal (INTADn) is generated after A/D conversion is started by the first generation of selection trigger 1 or 2. In the extension buffer mode, the error flag is set to 1 in accordance with a specified error condition if selection trigger 1 or 2, or selection load trigger 1 or 2 is generated during this period (except, however, the case in **Caution 2** in **12.4.11 (1) Error detection function**).

Remark n = 0, 1

12.6.7 Variation of A/D conversion results

The results of the A/D conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. To reduce the variation, take counteractive measures with the program, such as by averaging the A/D conversion results.

12.6.8 A/D conversion result hysteresis characteristics

Successive comparison type A/D converters hold an analog input voltage in an internal sample & hold capacitor and then perform A/D conversion. After the A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur if the output impedance from the analog input source is too high.

- When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value. Even if the conversion were to be performed at the same potential, the results may thus vary.
- When switching the analog input channel, hysteresis characteristics may appear where the conversion result is affected by the previous channel value. This is because one A/D converter is used for the A/D conversions. Even if the conversion were to be performed at the same potential, the results may thus vary.

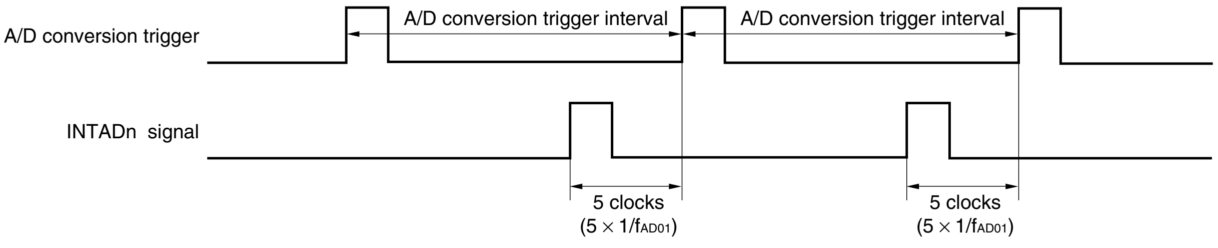
To obtain more accurate conversion results, lower the output impedance from the analog input source or execute A/D conversion twice consecutively on the same channel, and discard the first conversion result.

12.6.9 A/D conversion trigger interval for continuous conversion

For the A/D conversion trigger interval for continuous conversion, secure at least the minimum trigger interval shown below before inputting the next trigger. Otherwise, the trigger will be invalid (not retained).

Minimum trigger interval clock count = A/D conversion clock count + 5 clocks
Minimum trigger interval time = Minimum trigger interval clock count × 1/f_{AD01}

Example f_{AD01} = 10 MHz, A/D conversion time = 3.2 μs, A/D conversion clock count = 32 clocks
Minimum trigger interval clock count = 32 + 5 = 37
Minimum trigger interval timer = 37 × 1/10 = 3.7 [μs]



Remark n = 0, 1

12.7 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1 LSB (Least Significant Bit). The percentage of 1 LSB with respect to the full scale is expressed by %FSR (Full Scale Range). %FSR indicates the ratio of analog input voltage that can be converted as a percentage, and is always represented by the following formula regardless of the resolution.

1%FSR = (Max. value of analog input voltage that can be converted – Min. value of analog input voltage that can be converted)/100
= (AV_{REFPn} – 0)/100
= AV_{REFPn}/100

1 LSB is as follows when the resolution is 12 bits.

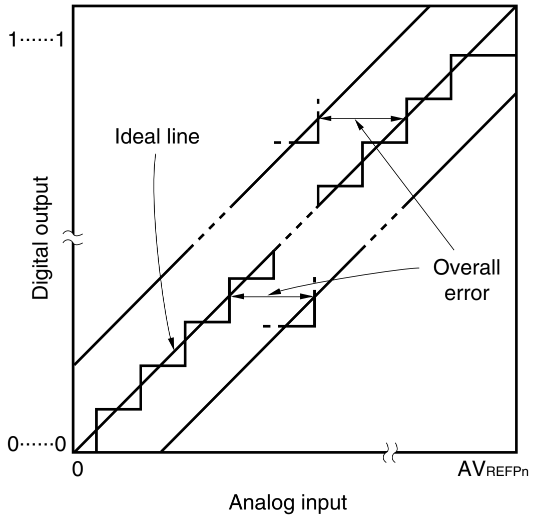
1 LSB = $1/2^{12}$ = 1/4,096
= 0.024%FSR

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, linearity error and errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

Figure 12-23. Overall Error

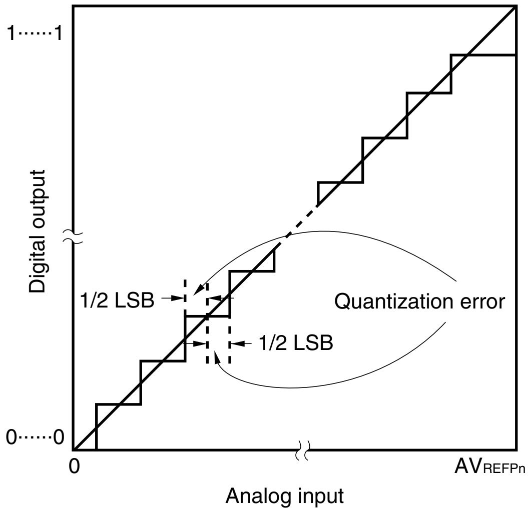


(3) Quantization error

When analog values are converted to digital values, a ±1/2 LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2 LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

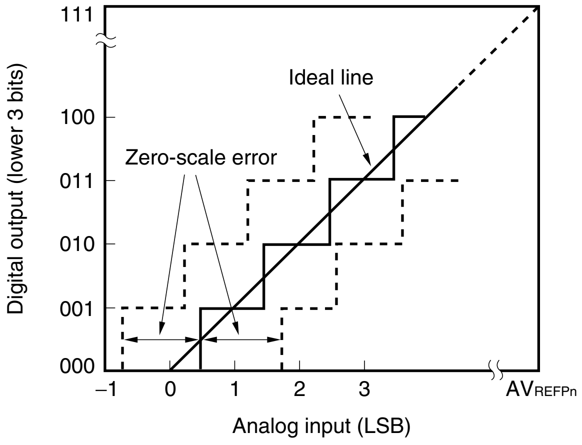
Figure 12-24. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0……000 to 0……001.

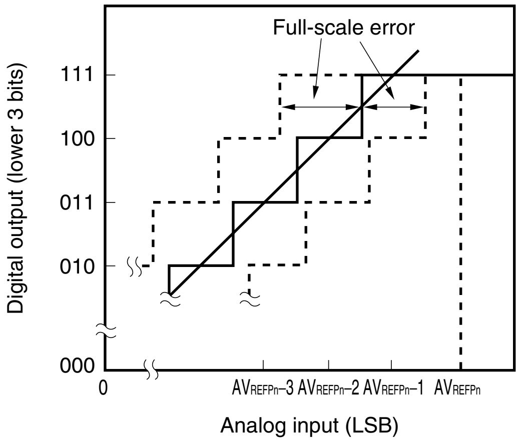
Figure 12-25. Zero-Scale Error



(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (full-scale value – 3/2 LSB) when the digital output changes from 1……110 to 1……111.

Figure 12-26. Full-Scale Error

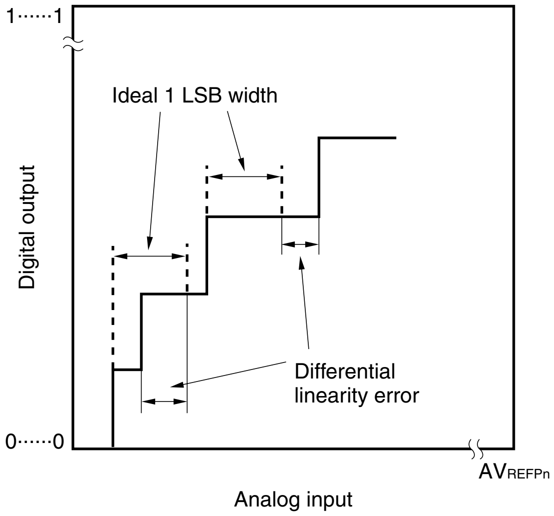


(6) Differential linearity error

While the ideal width of code output is 1 LSB, this indicates the difference between the actual measurement value and the ideal value.

This indicates the basic characteristics of the A/D conversion when the voltage applied to the analog input pins of the same channel is consistently increased bit by bit from AV_{SSn} to AV_{REFPn}. See **12.7 (2) Overall error** for when the input voltage is increased or decreased, or when two or more channels are used.

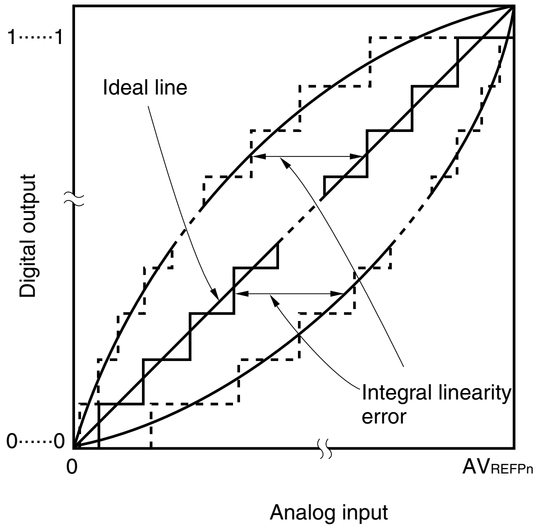
Figure 12-27. Differential Linearity Error



(7) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

Figure 12-28. Integral Linearity Error



(8) Conversion time

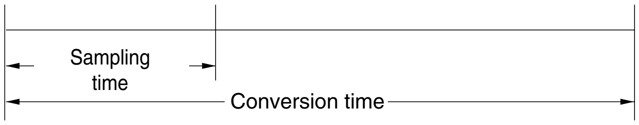
This expresses the time from when the trigger is generated to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.

Figure 12-29. Sampling Time



CHAPTER 13 A/D CONVERTER 2

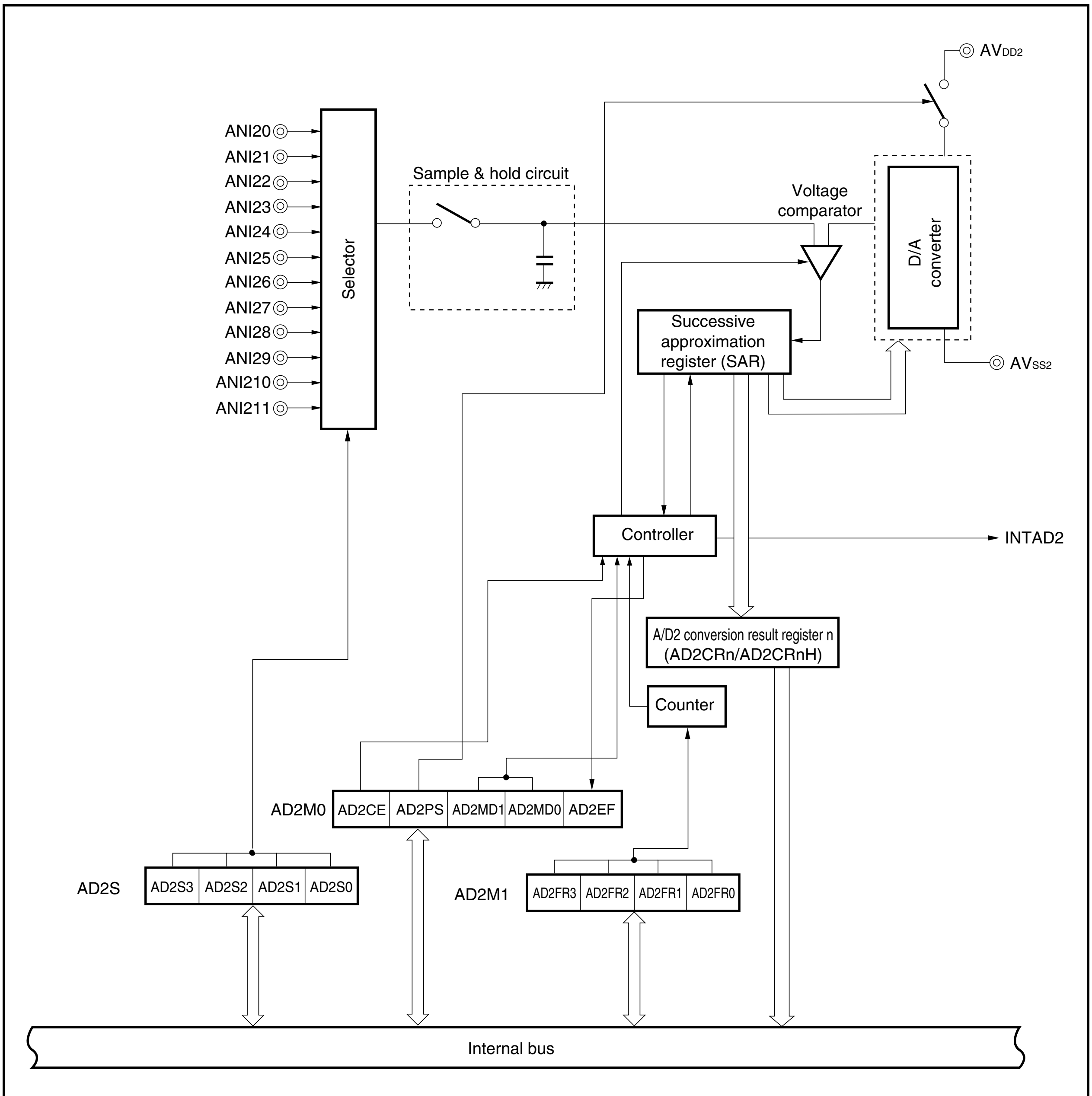
13.1 Features

- On-chip 10-bit resolution A/D converter
- Analog input
  ANI20 to ANI211 (12 channels)
- A/D conversion result register
  AD2CR0 to AD2CR11 (10 bits × 12)
- A/D conversion trigger mode
  Software trigger mode
- A/D conversion operation mode
  Continuous select mode
  Continuous scan mode
  One-shot select mode
  One-shot scan mode
- Successive comparison approximation method
- Operating voltage: EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} (V850E/IH4-H only) = AV_{DD2} = 4.0 to 5.5 V

13.2 Configuration

The block diagram is shown below.

Figure 13-1. Block Diagram of A/D Converter 2



Remark n = 0 to 11

Cautions 1. If there is noise at the analog input pin (ANI2n) and at the A/D converter power supply voltage pin (AV_{DD2}), that noise may generate an illegal conversion result.

Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.

An example of this software processing is shown below.

- **Take the average result of a number of A/D conversions and use that as the A/D conversion result.**
- **Execute a number of A/D conversions successively and use those results, omitting any exceptional results that may have been obtained.**
- **If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.**

2. Do not apply a voltage outside the AV_{SS2} to AV_{DD2} range to the pins that are used as input pins of A/D converter 2.

A/D converter 2 consists of the following hardware.

Table 13-1. Configuration of A/D Converter 2

Item	Configuration
Analog input	ANI20 to ANI211 (12 channels)
Registers	Successive approximation register (SAR) A/D2 conversion result registers 0 to 11 (AD2CR0 to AD2CR11) A/D2 conversion result registers 0H to 11H (AD2CR0H to AD2CR11H): Only the higher 8 bits can be read
Control registers	A/D converter 2 mode registers 0, 1 (AD2M0, AD2M1) A/D converter 2 channel specification register (AD2S)

(1) Successive approximation register (SAR)

The SAR register is a register that compares the voltage value of an analog input pin with the value of the voltage tap of the D/A converter and holds the result, starting from the most significant bit (MSB).

If data is held in the SAR all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred in AD2CRn register.

When all the specified A/D conversion operations have ended, an A/D2 conversion end interrupt request signal (INTAD2) is generated.

(2) A/D conversion result register n (AD2CRn), A/D conversion result register nH (AD2CRnH)

The AD2CRn register is a register that holds the A/D conversion results. The conversion result is stored in the higher 10 bits of the AD2CRn register corresponding to the analog input. The lower 6 bits of these registers are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the AD2CRn register.

To read the result of A/D conversion in 16-bit units, specify the AD2CRn register. To read the higher 8 bits, specify the AD2CRnH register.

Caution The contents of the AD2CRn register may become undefined depending on the operation to write the AD2M0, AD2M1, and AD2S registers. Read the result of conversion from the AD2CRn register after conversion and before writing the AD2M0, AD2M1, and AD2S registers. The correct conversion result cannot be read from the AD2CRn register if any other procedure is used.

(3) Sample & hold circuit

The sample & hold circuit samples the analog input signals selected by the input circuit and sends the sampled data to the voltage comparator. This circuit holds the sampled analog input voltage during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the value that is sampled and held with the voltage generated from the voltage tap of the D/A converter.

(5) D/A converter

The D/A converter is connected between AV_{DD2} and AV_{SS2} and generates a voltage to be compared with an input analog signal.

(6) ANI2n pin

The ANI2n pin is an analog input pin for A/D converter 2. This pin inputs the analog signals to be A/D converted. Pins other than the one that is selected by the AD2S register as analog signal input pins can be used as input port pins.

Cautions 1. Make sure that the voltages input to the ANI2n pin do not exceed the rated values. If a voltage higher than or equal to AV_{DD2} or lower than or equal to AV_{SS2} is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

2. The analog input pin (ANI2n) is alternately used as input port pin (P7n). If an instruction to input a signal to port 7 is executed during conversion when one of ANI2n is selected for A/D conversion, the resolution for conversion may drop.

(7) AV_{DD2} pin

The AV_{DD2} pin alternately functions as the pin for inputting the positive power supply and reference voltage of A/D converter 2. This pin converts signals input to the ANI2n pin to digital signals based on the voltage applied between AV_{DD2} and AV_{SS2}.

Always make the potential at this pin the same as that at the EV_{DD0}, EV_{DD1}, EV_{DD2}, and EV_{DD3} pins (V850E/IH4-H only) even when A/D converter 2 is not used.

The operating voltage range of the AV_{DD2} pin is $EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3}$ (V850E/IH4-H only) $= AV_{DD2} = 4.0$ to 5.5 V.

(8) AV_{SS2} pin

This is the ground pin of A/D converter 2. Always make the potential at this pin the same as that at the EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only) and EV_{SS4} pins even when A/D converter 2 is not used.

Remark n = 0 to 11

13.3 Control Registers

A/D converter 2 is controlled by the following registers.

- A/D converter 2 mode registers 0, 1 (AD2M0 to AD2M1)
- A/D converter 2 channel specification register (AD2S)

The following registers are also used.

- A/D2 conversion result register n (AD2CRn)
- A/D2 conversion result register nH (AD2CRnH)

(1) A/D converter 2 mode register 0 (AD2M0)

The AD2M0 register is a register that specifies the operation mode and controls conversion operations.

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFB80H

	<7>	6	5	4	3	2	1	0
AD2M0	AD2CE	AD2PS	AD2MD1	AD2MD0	0	0	0	AD2EF

AD2CE	Control of A/D conversion operation
0	Conversion operation stopped
1	Conversion operation enabled

AD2PS	A/D conversion control
0	A/D power on
1	A/D power off

- The first result of conversion by the A/D converter 2 becomes valid when the AD2CE bit is set to 1 (conversion is enabled) at least 2 μs after the AD2PS bit is set to 1 (A/D power is turned on).
  If the AD2CE bit is set to 1 before 2 μs pass, the conversion operation is started and ends after the A/D conversion time, but the conversion result is undefined.
- When the A/D converter 2 is not used, set to 0 the AD2CE bit (stop conversion operation) and the AD2PS bit (turn off A/D power) to reduce the power consumption.
- Do not set the AD2PS2 bit during A/D conversion operation (AD2EF bit = 1).
  While the A/D conversion operation is not performed, the AD2CE and AD2PS bits can be simultaneously cleared to 0.

AD2MD1	AD2MD0	Specification of operation mode
0	0	Successive select mode
0	1	Successive scan mode
1	0	One-shot select mode
1	1	One-shot scan mode

AD2EF	Status of A/D converter 2 (status)
0	During A/D conversion stop
1	During A/D conversion operation

Cautions 1. Writing to bit 0 is ignored.

2. The conversion resolution of the pin to which an analog signal is input first immediately after A/D conversion is started may drop. For details, see 13.7 (6) About AV_{DD2} pin.

3. A/D conversion is stopped and started again from the beginning if the AD2M0 and AD2S registers are written during A/D conversion operation (AD2EF bit = 1).

4. Be sure to set bits 1 to 3 to “0”.

(2) A/D converter 2 mode register 1 (AD2M1)

The AD2M1 register is a register that specifies the number of A/D conversion clocks and A/D conversion time.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFB81H

	7	6	5	4	3	2	1	0
AD2M1	0	0	0	0	AD2FR3	AD2FR2	AD2FR1	AD2FR0

Cautions 1. See Table 13-2 Setting Example During Conversion Mode for the AD2FR3 to AD2FR0 bits.

2. Changing the AD2FR3 to AD2FR0 bits is prohibited during conversion operation (AD2CE bit = 1).

3. Be sure to set bits 4 to 7 to “0”.

Table 13-2. Setting Example During Conversion Mode

AD2FR3	AD2FR2	AD2FR1	AD2FR0	Number of A/D Conversion Clocks[Note]	A/D Conversion Time	f_{AD2} = 50 MHz (f_{XX} = 100 MHz)	f_{AD2} = 40 MHz (f_{XX} = 80 MHz)
0	0	1	1	124	124/f_{AD2}	Setting prohibited	3.10 μs
0	1	0	0	155	155/f_{AD2}	3.10 μs	3.88 μs
0	1	0	1	186	186/f_{AD2}	3.72 μs	4.65 μs
0	1	1	0	217	217/f_{AD2}	4.34 μs	5.43 μs
0	1	1	1	248	248/f_{AD2}	4.96 μs	6.20 μs
1	0	0	0	279	279f_{AD2}	5.58 μs	6.98 μs
1	0	0	1	310	310/f_{AD2}	6.20 μs	7.75 μs
1	0	1	0	341	341/f_{AD2}	6.82 μs	8.53 μs
1	0	1	1	372	372/f_{AD2}	7.44 μs	9.30 μs
1	1	0	0	403	403/f_{AD2}	8.06 μs	Setting prohibited
1	1	0	1	434	434/f_{AD2}	8.68 μs	Setting prohibited
1	1	1	0	465	434/f_{AD2}	9.30 μs	Setting prohibited
1	1	1	1	496	496/f_{AD2}	9.92 μs	Setting prohibited
Other than above					Setting prohibited		

Note The number of clocks (f_{AD2}) from the start to the end of A/D conversion.

Caution Set the A/D conversion time in a range from 3.00 to 10.00 μs.

Remark f_{AD2}: Operating clock of A/D converter 2

(3) A/D converter 2 channel specification register (AD2S)

The AD2S register is a register that specifies the analog input pin to be A/D-converted.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFB82H

	7	6	5	4	3	2	1	0
AD2S	0	0	0	0	AD2S3	AD2S2	AD2S1	AD2S0

AD2S3	AD2S1	AD2S0	AD2S0	Select mode	Scan mode
0	0	0	0	ANI20	ANI20
0	0	0	1	ANI21	ANI20, ANI21
0	0	1	0	ANI22	ANI20 to ANI22
0	0	1	1	ANI23	ANI20 to ANI23
0	1	0	0	ANI24	ANI20 to ANI24
0	1	0	1	ANI25	ANI20 to ANI25
0	1	1	0	ANI26	ANI20 to ANI26
0	1	1	1	ANI27	ANI20 to ANI27
1	0	0	0	ANI28	ANI20 to ANI28
1	0	0	1	ANI29	ANI20 to ANI29
1	0	1	0	ANI210	ANI20 to ANI210
1	0	1	1	ANI211	ANI20 to ANI211
Other than above				Setting prohibited	

Caution Be sure to set bits 4 to 7 to “0”.

(4) A/D2 conversion result registers n, nH (AD2CRn, AD2CRnH)

The AD2CRn and AD2CRnH registers are registers that hold the A/D conversion results. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 10 bits of the AD2CRn register. The lower 6 bits of these registers are always 0 when read.

The higher 8 bits of A/D conversion result are read to the AD2CRnH register.

These registers can only be read in 16-bit or 8-bit units. When the A/D conversion results are read in 16-bit units, the AD2CRn register is specified, and when the higher 8 bits are read, the AD2CRnH register is specified.

Reset sets AD2CRn register to 0000H and AD2CRnH register to 00H.

Caution If a write operation is performed on the AD2M0, AD2M1, and AD2S registers, the contents of the AD2CRn register may become undefined. Read the conversion result after the conversion operation and before performing a write operation on the AD2M0, AD2M1, and AD2S registers. The correct conversion result may not be read if the timing is other than the above.

After reset: 0000H R Address: AD2CR0 FFFFFB90H, AD2CR1 FFFFFB92H,
AD2CR2 FFFFFB94H, AD2CR3 FFFFFB96H,
AD2CR4 FFFFFB98H, AD2CR5 FFFFFB9AH,
AD2CR6 FFFFFB9CH, AD2CR7 FFFFFB9EH,
AD2CR8 FFFFFBA0H, AD2CR9 FFFFFBA2H,
AD2CR10 FFFFFBA4H, AD2CR11 FFFFFBA6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD2CRn (n = 0 to 11)	AD 29	AD 28	AD 27	AD 26	AD 25	AD 24	AD 23	AD 22	AD 21	AD 20	0	0	0	0	0	0

After reset: 00H R Address: AD2CR0H FFFFFB91H, AD2CR1H FFFFFB93H,
AD2CR2H FFFFFB95H, AD2CR3H FFFFFB97H,
AD2CR4H FFFFFB99H, AD2CR5H FFFFFB9BH,
AD2CR6H FFFFFB9DH, AD2CR7H FFFFFB9FH,
AD2CR8H FFFFFBA1H, AD2CR9H FFFFFBA3H,
AD2CR10H FFFFFBA5H, AD2CR11H FFFFFBA7H

	7	6	5	4	3	2	1	0
AD2CRnH (n = 0 to 11)	AD29	AD28	AD27	AD26	AD25	AD24	AD23	AD22

The correspondence between the analog input pins and the AD2CRn and AD2CRnH registers is shown below.

Table 13-3. Correspondence Between Analog Input Pins and AD2CRn and AD2CRnH Registers

Analog Input Pin	A/D Conversion Result Register
ANI20	AD2CR0, AD2CR0H
ANI21	AD2CR1, AD2CR1H
ANI22	AD2CR2, AD2CR2H
ANI23	AD2CR3, AD2CR3H
ANI24	AD2CR4, AD2CR4H
ANI25	AD2CR5, AD2CR5H
ANI26	AD2CR6, AD2CR6H
ANI27	AD2CR7, AD2CR7H
ANI28	AD2CR8, AD2CR8H
ANI29	AD2CR9, AD2CR9H
ANI210	AD2CR10, AD2CR10H
ANI211	AD2CR11, AD2CR11H

The relationship between the analog voltage input to the analog input pin (ANI2n) and the A/D conversion result (of A/D2 conversion result register n (AD2CRn)) is as follows:

$$\text{SAR} = \text{INT}\left(\frac{V_{IN}}{AV_{DD2}} \times 1{,}024 + 0.5\right)$$

$$\text{ADCR}^{\text{Note}} = \text{SAR} \times 64$$

or,

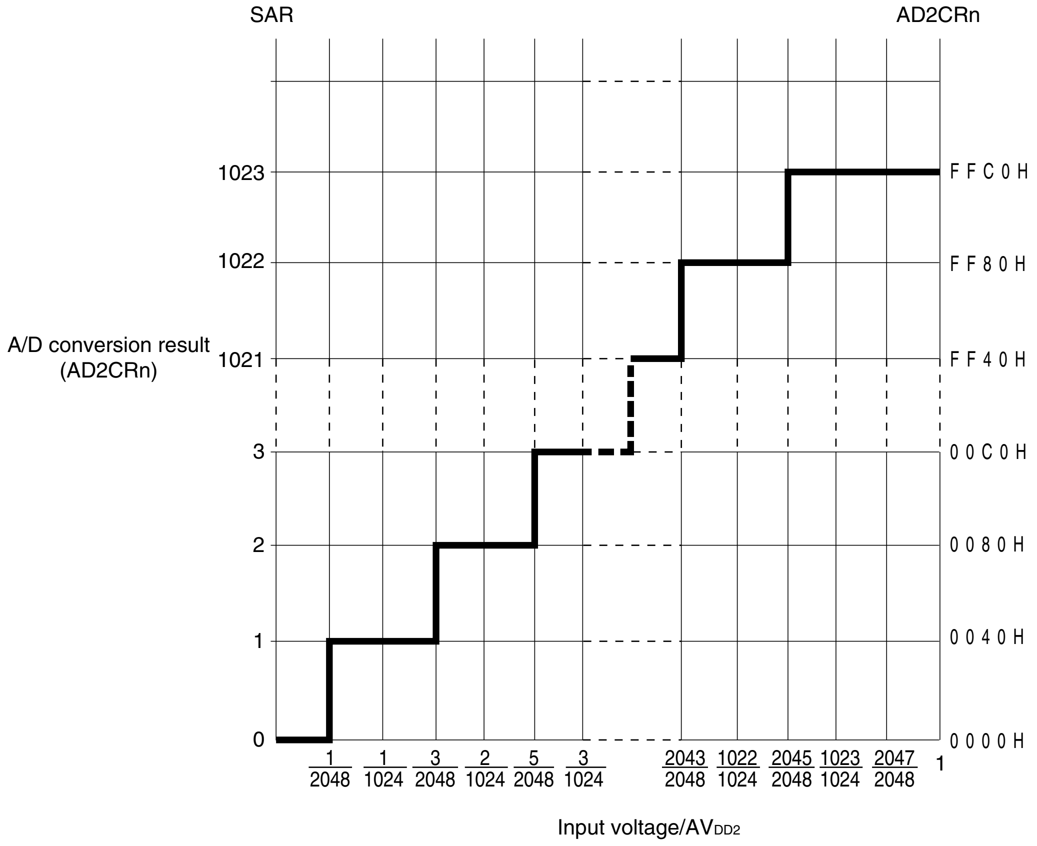
$$(\text{SAR} - 0.5) \times \frac{AV_{DD2}}{1{,}024} \le V_{IN} < (\text{SAR} + 0.5) \times \frac{AV_{DD2}}{1{,}024}$$

INT( ): Function that returns the integer of the value in ( )
V_{IN}: Analog input voltage
AV_{DD2}: AV_{DD2} pin voltage
ADCR: Value of A/D2 conversion result register n (AD2CRn)

Note The lower 6 bits of the AD2CRn register are fixed to 0.

The relationship between the analog input voltage and the A/D conversion results is shown in Figure 13-2.

Figure 13-2. Relationship Between Analog Input Voltage and A/D Conversion Results



Remark n = 0 to 11

13.4 Operation

13.4.1 Basic operation

<1> Set the AD2M0.AD2PS bit to 1 to turn on A/D power while the AD2M0.AD2CE bit = 0.
At this time, bits other than the AD2M0.AD2CE bit can be simultaneously set.

<2> Select an operation mode of A/D conversion and A/D conversion time by using the AD2M0, AD2M1, and AD2S registers.

<3> Setting the AD2M0.AD2CE bit to 1 (enable conversion) at least 2 μs after turning on A/D power (AD2M0.AD2PS bit = 0 → 1) starts A/D conversion.
If the AD2CE bit is set to 1 before 2 μs passes, the conversion operation is started and ends after A/D conversion time, but the conversion result is undefined.

<4> When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<5> When sampling has been performed for a specific time, the sample & hold circuit enters the hold status, and holds the input analog voltage until A/D conversion ends.

<6> Set bit 9 of the successive approximation register (SAR) and changes the level of the voltage tap of the D/A converter to the reference voltage ($1/2AV_{DD2}$).

<7> The voltage generated by the voltage tap of the D/A converter is compared with the analog input voltage by a voltage comparator. If the analog input voltage is found to be greater than ($1/2AV_{DD2}$) as a result of comparison, the MSB of the SAR register remains set. If the analog input voltage is less than ($1/2AV_{DD2}$), the MSB is reset.

<8> Next, bit 8 of the SAR register is automatically set, and the next comparison is started. The voltage tap of the D/A converter is selected according to the value of bit 9, to which the result has been already set as shown below.

Bit 9 = 1: ($3/4AV_{DD2}$)
Bit 9 = 0: ($1/4AV_{DD2}$)

The voltage tap of the D/A converter and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison as shown below.

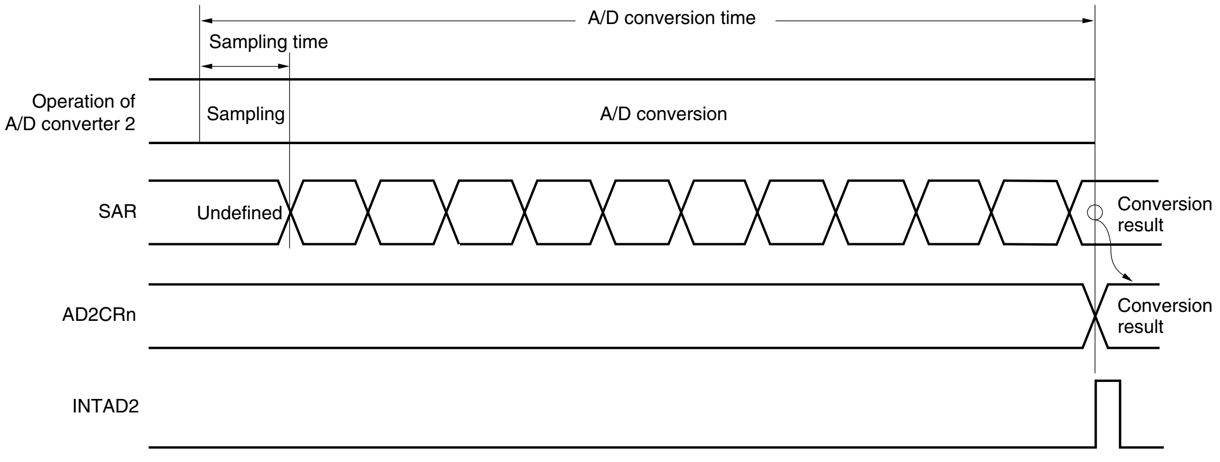
Analog input voltage ≥ Voltage tap of D/A converter: Bit 8 = 1
Analog input voltage ≤ Voltage tap of D/A converter: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

<9> When comparison of 10 bits has been completed, the valid digital result remains in the SAR register. This value is transferred to the AD2CRn register and the conversion result is stored in this register (n = 0 to 11). An A/D2 conversion end interrupt request signal (INTAD2) is generated simultaneously in the select mode and when all the specified A/D conversion operations are completed in the scan mode.

<10> In the continuous select mode or continuous scan mode, <4> to <9> are repeated unless the AD2CE bit is set to 0 after completion of A/D conversion.
In the one-shot select mode or one-shot scan mode, the conversion operation is stopped after it is completed (at this time, the AD2M0.AD2CE bit holds 1 and is not automatically cleared). Write 1 to the AD2CE bit to start conversion operation again.

Figure 13-3. Basic Operation of A/D Converter 2



Remark n = 0 to 11

13.4.2 Trigger mode

Trigger mode that serve as the start timing of an A/D conversion operation is software trigger mode.

This mode is set by the AD2M0 register.

(1) Software trigger mode

In this mode, the analog input pin (ANI2n) specified by the AD2S.AD2S3 to AD2S.AD2S0 bits is used for the A/D conversion start timing by setting the AD2M0.AD2CE bit to 1.

After A/D conversion ends, the conversion result is stored in A/D2 conversion result register n (AD2CRn).

An A/D2 conversion end interrupt request signal (INTAD2) is generated simultaneously when A/D conversion operations are completed in the select mode.

INTAD2 interrupt request signal is generated and when all the specified A/D conversion operations are completed in the scan mode.

If the operation mode set by the AD2M0.AD2MD1 and AD2M0.AD2MD0 bits is the continuous select mode or continuous scan mode, the conversion operation is repeated unless the AD2M0.AD2CE bit is set to 0. In the one-shot select mode or one-shot scan mode, the conversion operation is stopped after A/D conversion ends.

The AD2M0.AD2EF bit is set to 1 (conversion in progress) when A/D conversion is started, and set to 0 (conversion stops) when it is completed.

If the AD2M0 and AD2S registers are written during A/D conversion, the conversion is stopped and executed again from the beginning.

Remark n = 0 to 11

13.4.3 Operation mode

There are four operation modes to which the ANI2n pin is set: continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode. These modes are set by the AD2M0.AD2MD1 and AD2M0.AD2MD0 registers.

The relationship between the AD2M0, AD2M1, and AD2S registers and operation mode is shown below.

Trigger Mode	Operation Mode	Set Value		
		AD2M0	AD2M1	AD2S
Software trigger	Continuous select	X100000XB	0000XXXXB	0000XXXXB
	Continuous scan	X101000XB	0000XXXXB	0000XXXXB
	One-shot select	X110000XB	0000XXXXB	0000XXXXB
	One-shot scan	X111000XB	0000XXXXB	0000XXXXB

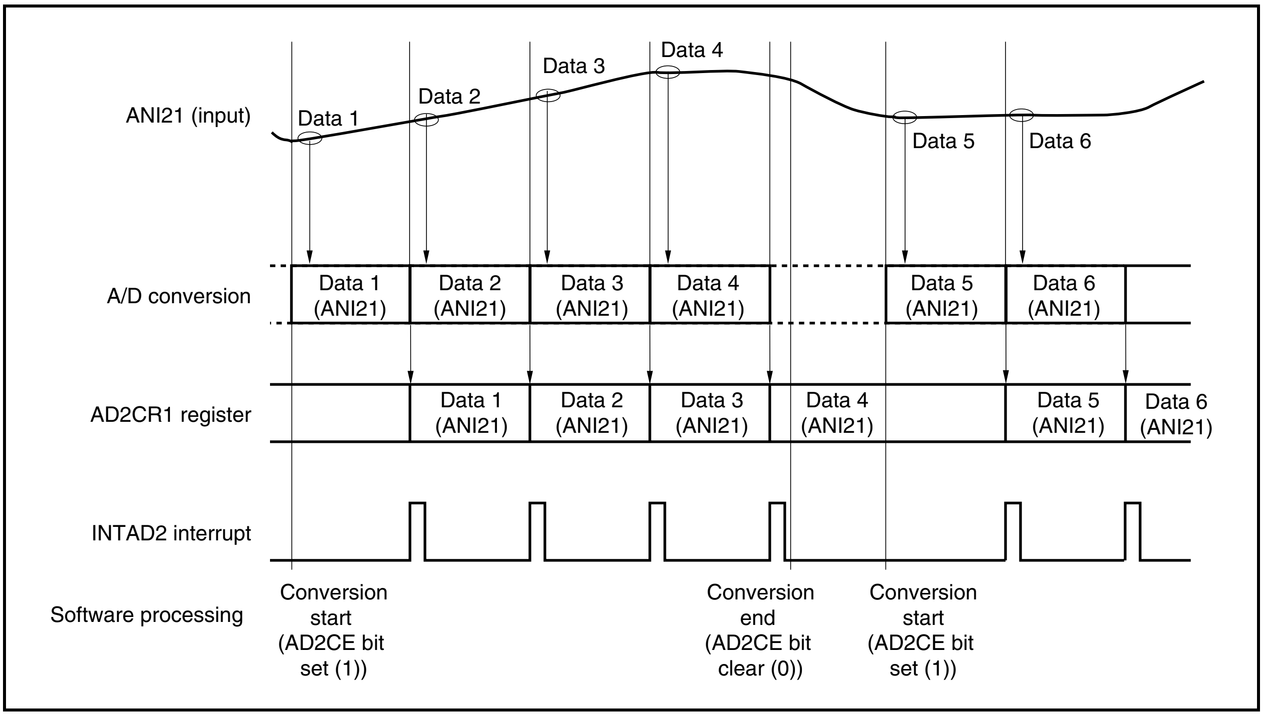
(1) Continuous select mode

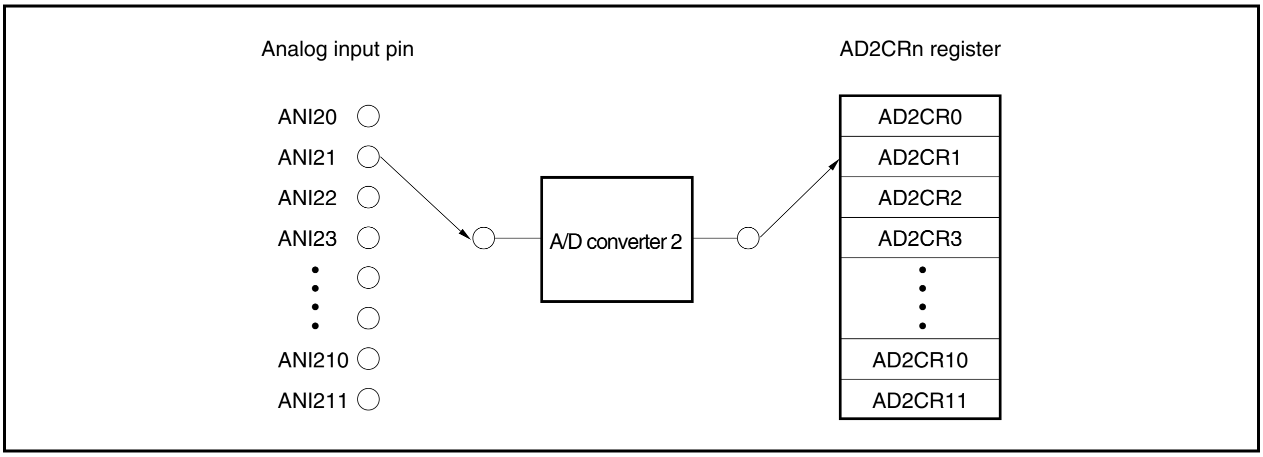
In this mode, the analog input pin (ANI2n) specified by the AD2S register is A/D-converted continuously. The conversion results are stored in the AD2CRn register corresponding to the ANI2n pin. The ANI2n pin and the AD2CRn register correspond one to one, and an A/D2 conversion end interrupt request signal (INTAD2) is generated each time one A/D conversion ends.

After A/D conversion ends, the conversion is repeated again unless the AD2M0.AD2CE bit is set to 0.

Remark n = 0 to 11

Figure 13-4. Continuous Select Mode Operation Timing (When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 00, AD2S.AD2S3 to AD2S.AD2S0 Bits = 0001)



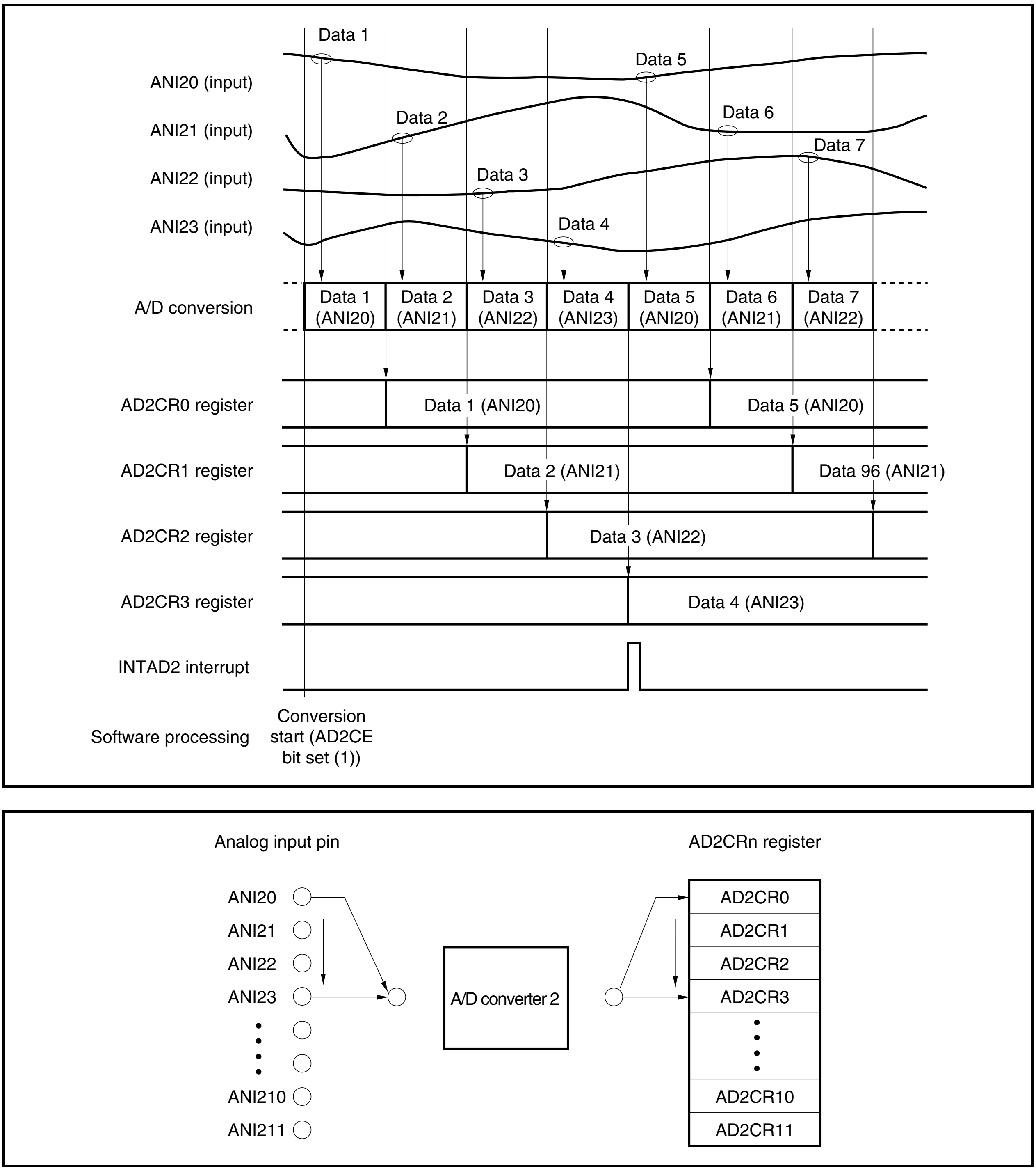


(2) Continuous scan mode

In this mode, the analog input pin (ANI2n) specified by the AD2S register is selected sequentially from the ANI20 pin, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin. When conversion of all the specified analog input pin ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion is started again from the ANI20 pin, unless the AD2M0.AD2CE bit is set to 0.

Remark n = 0 to 11

Figure 13-5. Continuous Scan Mode Operation Timing
(When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 01, AD2S.AD2S3 to AD2S.AD2S0 Bits = 0011)



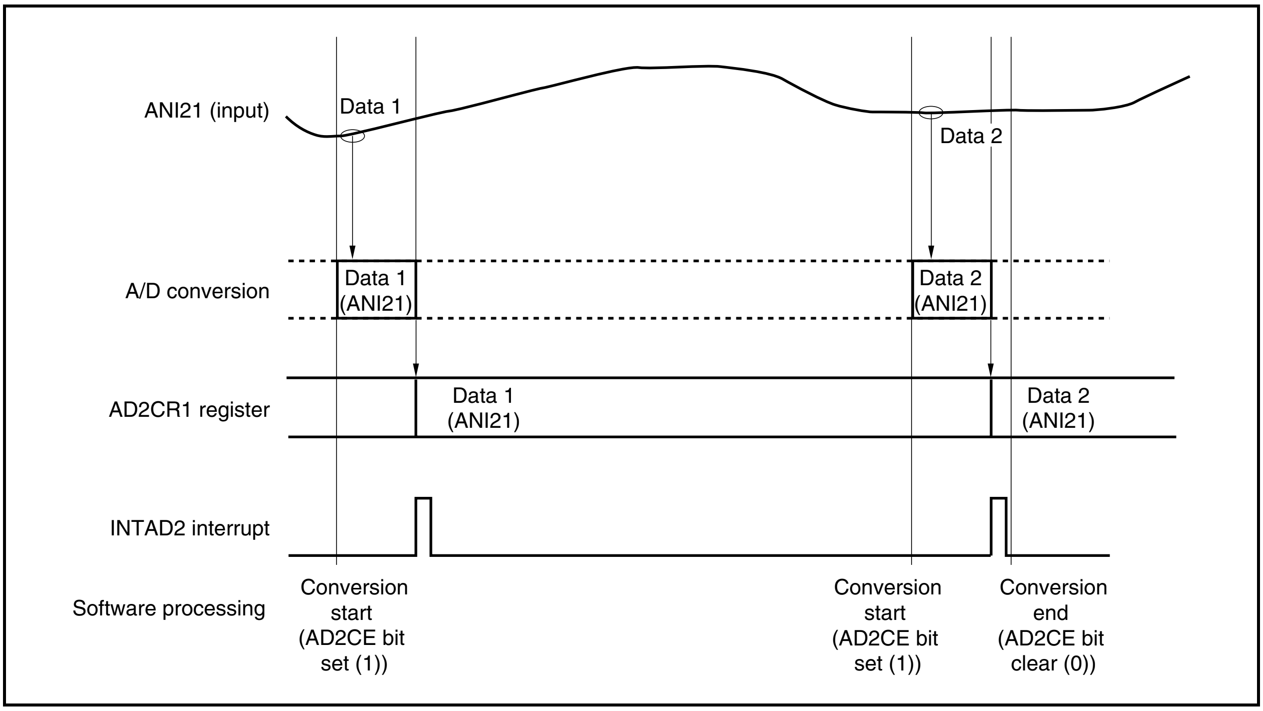
(3) One-shot select mode

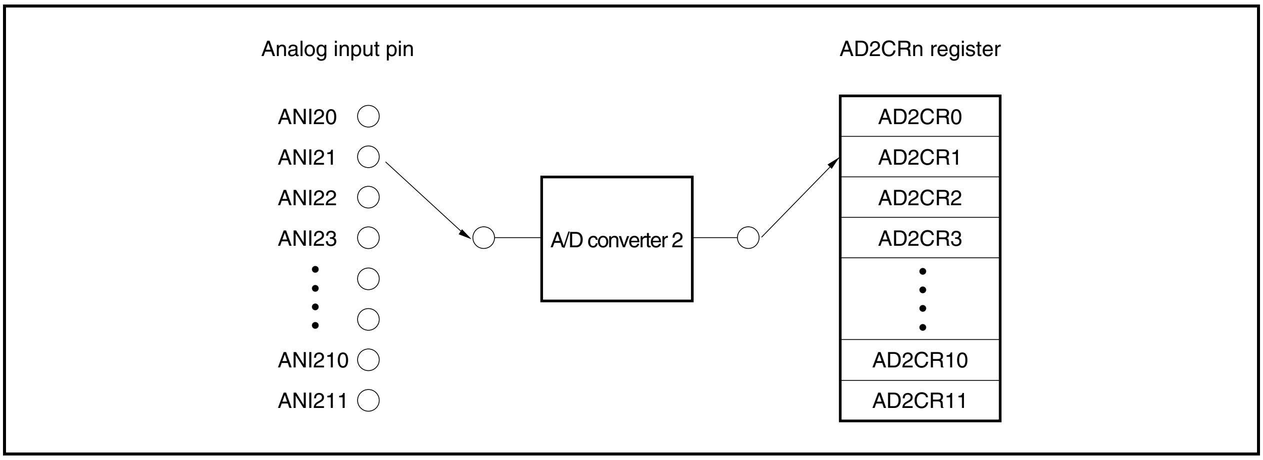
In this mode, the analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion result is stored in the AD2CRn register corresponding to the ANI2n pin. The ANI2n pin and the AD2CRn register correspond one to one, and an A/D2 conversion end interrupt request signal (INTAD2) is generated each time one A/D conversion ends.

After A/D conversion ends, the conversion operation is stopped.

Remark n = 0 to 11

Figure 13-6. One-Shot Select Mode Operation Timing
(When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 10, AD2S.AD2S3 to AD2S.AD2S0 Bits = 0001)



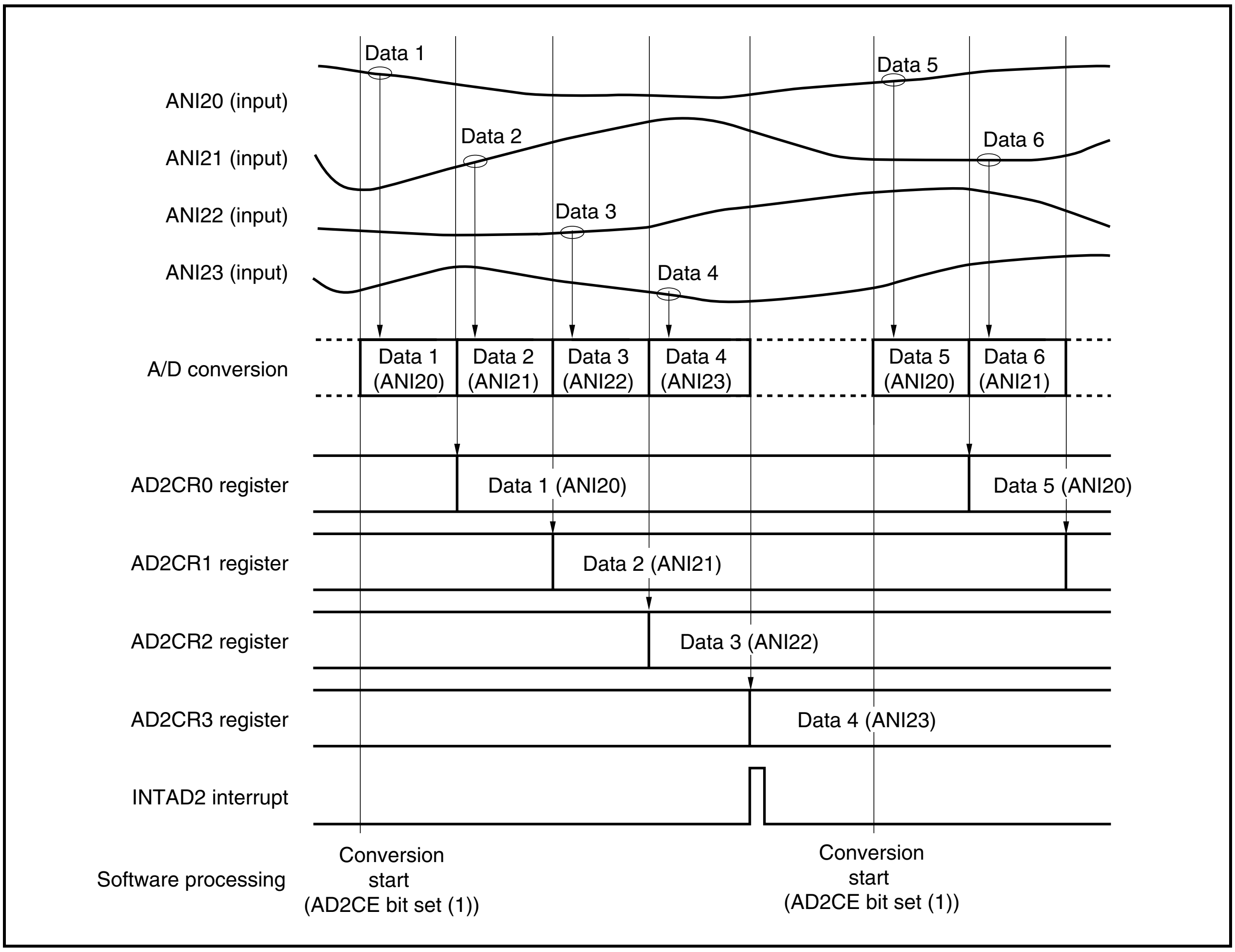


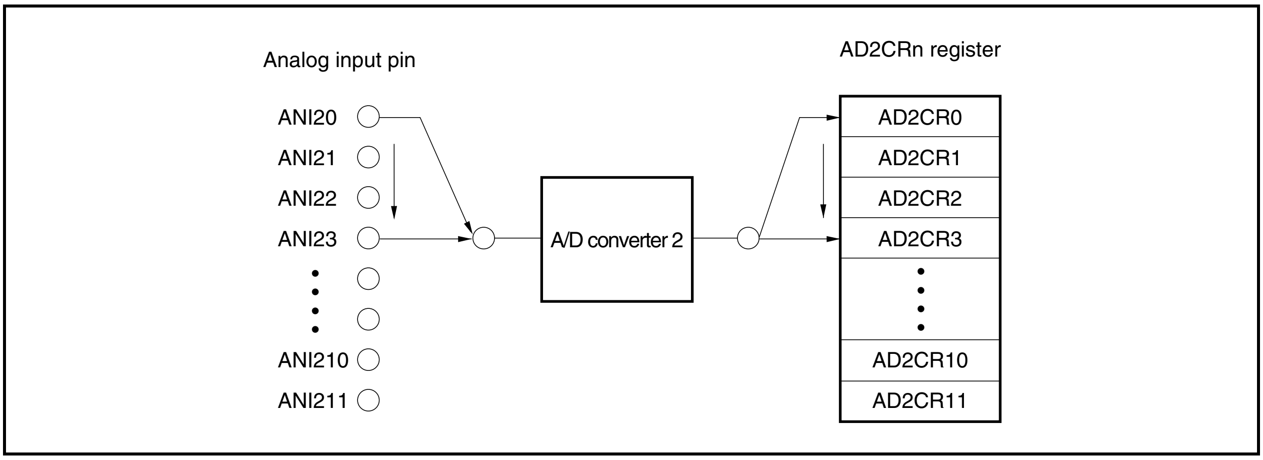
(4) One-shot scan mode

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin. When conversion of all the specified analog input pins ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion operation is stopped.

Remark n = 0 to 11

Figure 13-7. One-Shot Scan Mode Operation Timing (When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 11, AD2S.AD2S3 to AD2S.AD2S0 Bits = 0011)





13.5 Operation in Software Trigger Mode

When the AD2M0.AD2CE bit is set to 1, A/D conversion is started.

When A/D conversion is started, the AD2M0.AD2EF bit = 1 (conversion in progress).

If the AD2M0 and AD2S registers are written during A/D conversion, the conversion is stopped and executed again from the beginning.

(1) Operation in software trigger continuous select mode

In this mode, one analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion results are stored in one AD2CRn register. The ANI2n pin and AD2CRn register correspond one to one.

Each time an A/D conversion is executed, an A/D2 conversion end interrupt request signal (INTAD2) is generated and A/D conversion ends. After A/D conversion ends, the conversion is repeated again unless the AD2M0.AD2CE bit is set to 0.

It is not necessary to set (1) the AD2M0.AD2CE bit to restart A/D conversion[Note].

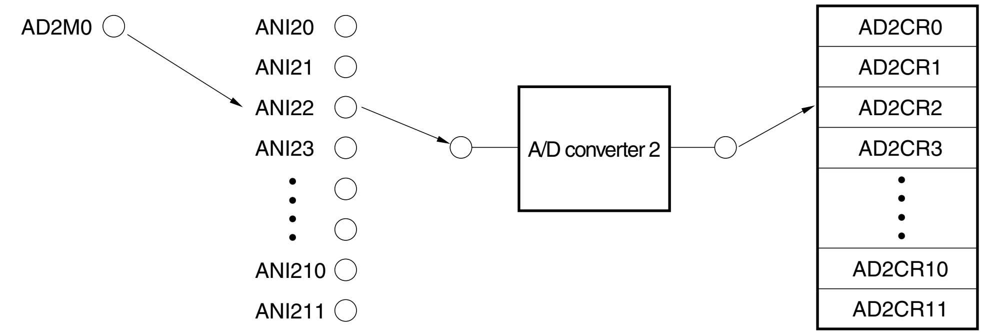
Note In the software trigger continuous select mode, the A/D conversion operation is not stopped unless the AD2M0.AD2CE bit is set to 0. If the AD2CRn register is not read before the next A/D conversion ends, it is overwritten.

This mode is suitable for applications in which the A/D conversion value of one analog input pin is read.

Analog Input Pin	A/D Conversion Result Register
ANI2n	AD2CRn

Remark n = 0 to 11

Figure 13-8. Operation Example of Software Trigger Continuous Select Mode



(1) The AD2CE bit = 1 (enable)
(2) The ANI22 pin is A/D-converted
(3) The conversion result is stored in the AD2CR2 register
(4) The AD2M0.AD2EF bit = 0
(5) The INTAD2 interrupt request signal is generated
(6) Return to (2)
(7) To end the conversion, the AD2CE bit = 0 (stop)

Remark This is an operation example with the following setting.
AD2M0.AD2MD1 and AD2M0.AD2MD0 bits = 00, AD2S.AD2S3 to AD2S.AD2S0 bits = 0010

(2) Software trigger continuous scan mode operations

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin.

When conversion of all the specified analog input pins ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion is started again from the ANI20 pin, unless the AD2M0.AD2CE bit is set to 0.

It is not necessary to set (1) the AD2M0.AD2CE bit to restart A/D conversion[Note].

Note In the software trigger continuous scan mode, the A/D conversion operation is not stopped unless the AD2M0.AD2CE bit is set to 0. If the AD2CRn register is not read before the next A/D conversion ends, it is overwritten.

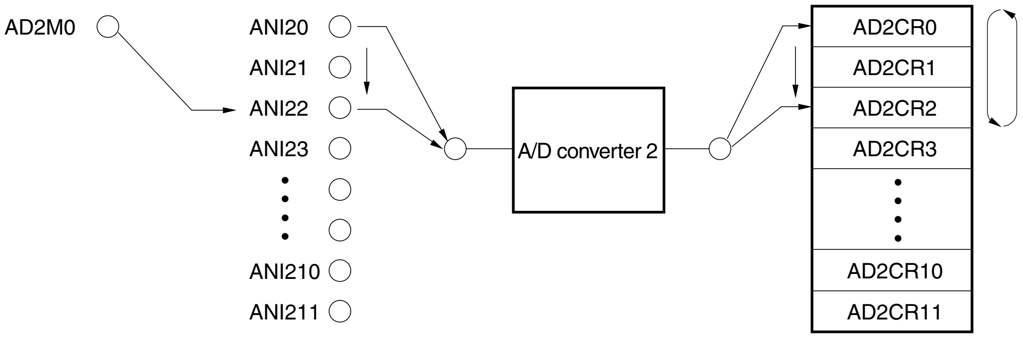
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

Analog Input Pin	A/D Conversion Result Register
ANI20	AD2CR0
:	:
ANI2n[Note]	AD2CRn

Note Set by the AD2S.AD2S0 to AD2S.AD2S3 bits.

Remark n = 0 to 11

Figure 13-9. Operation Example of Software Trigger Continuous Scan Mode



(1) The AD2CE bit = 1 (enable)
(2) The ANI20 pin is A/D-converted
(3) The conversion result is stored in the AD2CR0 register
(4) The ANI21 pin is A/D-converted
(5) The conversion result is stored in the AD2CR1 register
(6) The ANI22 pin is A/D-converted
(7) The conversion result is stored in the AD2CR2 register
(8) The INTAD2 interrupt request signal is generated
(9) Return to (2)
(10) To end the conversion, the AD2CE bit = 0 (stop)

Remark This is an operation example with the following setting.
AD2M0.AD2MD1 and AD2M0.AD2MD0 bits = 01, AD2S.AD2S3 to AD2S.AD2S0 bits = 0010

(3) Software trigger one-shot select mode

In this mode, the voltage of one analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion result is stored in one AD2CRn register. The ANI2n pin and the AD2CRn register correspond one to one.

Each time an A/D conversion is executed, an A/D2 conversion end interrupt request signal (INTAD2) is generated and A/D conversion ends. After A/D conversion ends, the conversion operation is stopped.

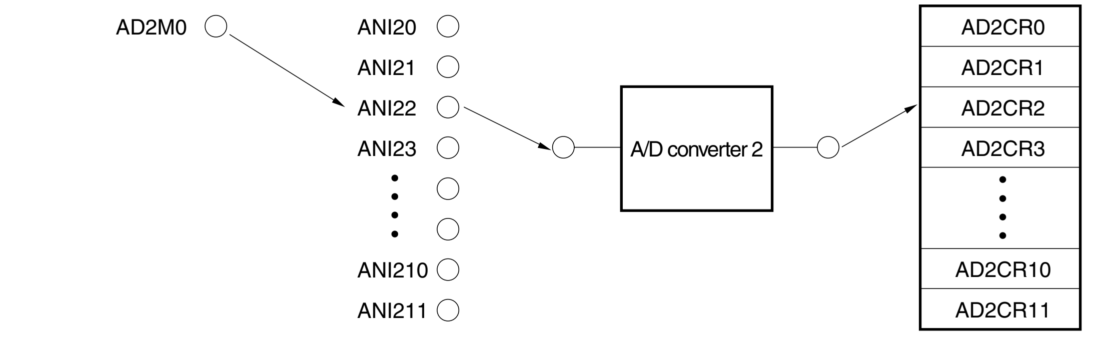
If the AD2M0.AD2CE bit is set to 1, A/D conversion can be restarted.

This mode is suitable for applications in which the results of each first-time A/D conversion are read.

Analog Input Pin	A/D Conversion Result Register
ANI2n	AD2CRn

Remark n = 0 to 11

Figure 13-10. Operation Example of Software Trigger One-Shot Select Mode



(1) The AD2CE bit = 1 (enable)

(2) The ANI22 pin is A/D-converted

(3) The conversion result is stored in the AD2CR2 register

(4) The AD2M0.AD2EF bit = 0

(5) The INTAD2 interrupt request signal is generated

Remark This is an operation example with the following setting.
AD2M0.AD2MD1 and AD2M0.AD2MD0 bits = 10, AD2S.AD2S3 to AD2S.AD2S0 bits = 0010

(4) Software trigger one-shot scan mode operations

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin.

When conversion of all the specified analog input pin ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion operation is stopped.

If the AD2M0.AD2CE bit is set to 1, A/D conversion can be restarted.

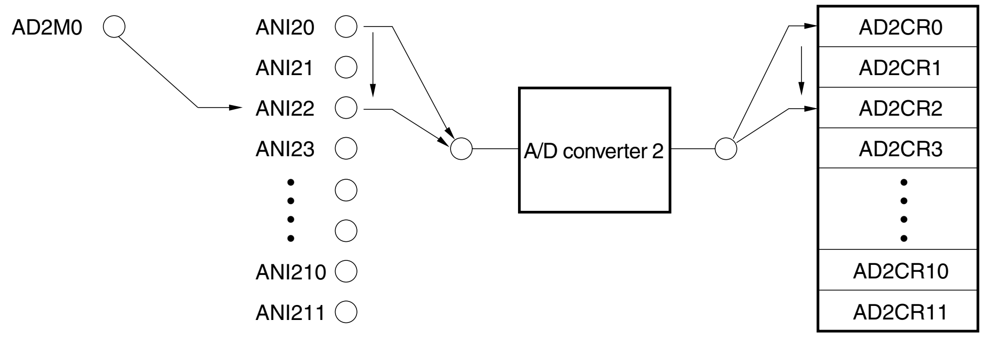
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

Analog Input Pin	A/D Conversion Result Register
ANI20	AD2CR0
⋮	⋮
ANI2n[Note]	AD2CRn

Note Set by the AD2S.AD2S0 to AD2S.AD2S3 bits.

Remark n = 0 to 11

Figure 13-11. Operation Example of Software Trigger One-Shot Scan Mode



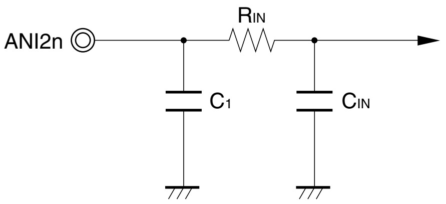
(1) The AD2CE bit = 1 (enable)
(2) The ANI20 pin is A/D-converted
(3) The conversion result is stored in the AD2CR0 register
(4) The ANI21 pin is A/D-converted
(5) The conversion result is stored in the AD2CR1 register
(6) The ANI22 pin is A/D-converted
(7) The conversion result is stored in the AD2CR2 register
(8) The AD2M0.AD2EF bit = 0
(9) The INTAD2 interrupt request signal is generated

Remark This is an operation example with the following setting.
AD2M0.AD2MD1 and AD2M0.AD2MD0 bits = 11, AD2S.AD2S3 to AD2S.AD2S0 bits = 0010

<R>

13.6 Internal Equivalent Circuit

The following figure shows the equivalent circuit of the analog input block.



R	C_1	C_2
2.6 kΩ	15 pF	6.2 pF

Remarks 1. The maximum values are shown (reference values).

2. n = 0 to 11

AD2M1 register				Number of A/D conversion clocks (f_{AD2})	Number of Sampling clocks (f_{AD2})
AD2FR3 bit	AD2FR2 bit	AD2FR1 bit	AD2FR0 bit		
0	0	1	1	124	66
0	1	0	0	155	82.5
0	1	0	1	186	99
0	1	1	0	217	115.5
0	1	1	1	248	132
1	0	0	0	279	148.5
1	0	0	1	310	165
1	0	1	0	341	181.5
1	0	1	1	372	198
1	1	0	0	403	214.5
1	1	0	1	434	231
1	1	1	0	465	247.5
1	1	1	1	496	264

13.7 Cautions

(1) When A/D converter is not used

When the A/D converter is not used, the power consumption can be reduced by clearing the AD2M0.AD2CE and AD2M0.AD2PS bits to 0.

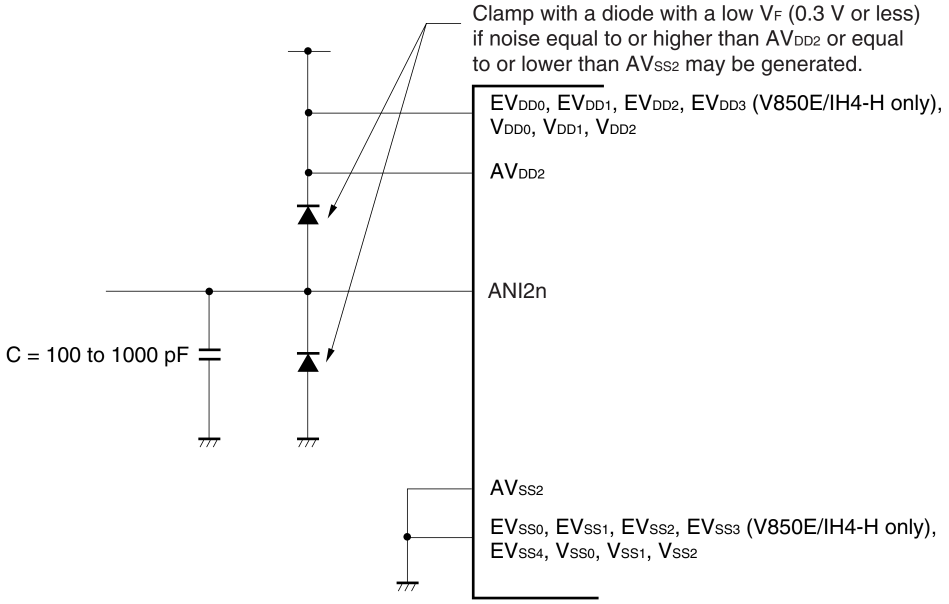
(2) Input range of ANI2n pin

Input the voltage within the specified range to the ANI2n pin. If a voltage equal to or higher than AV$_{DD2}$ or equal to or lower than AV$_{SS2}$ (even within the range of the absolute maximum ratings) is input to this pin, the conversion value of that channel is undefined, and the conversion value of the other channels may also be affected.

(3) Countermeasures against noise

To maintain the 10-bit resolution, the ANI2n pin must be effectively protected from noise. The influence of noise increases as the output impedance of the analog input source becomes higher. To lower the noise, connecting an external capacitor as shown in Figure 13-12 is recommended.

Figure 13-12. Processing of Analog Input Pin


Clamp with a diode with a low V$_F$ (0.3 V or less) if noise equal to or higher than AV$_{DD2}$ or equal to or lower than AV$_{SS2}$ may be generated.
EV$_{DD0}$, EV$_{DD1}$, EV$_{DD2}$, EV$_{DD3}$ (V850E/IH4-H only), V$_{DD0}$, V$_{DD1}$, V$_{DD2}$
AV$_{DD2}$
ANI2n
C = 100 to 1000 pF
AV$_{SS2}$
EV$_{SS0}$, EV$_{SS1}$, EV$_{SS2}$, EV$_{SS3}$ (V850E/IH4-H only), EV$_{SS4}$, V$_{SS0}$, V$_{SS1}$, V$_{SS2}$

Remark n = 0 to 11

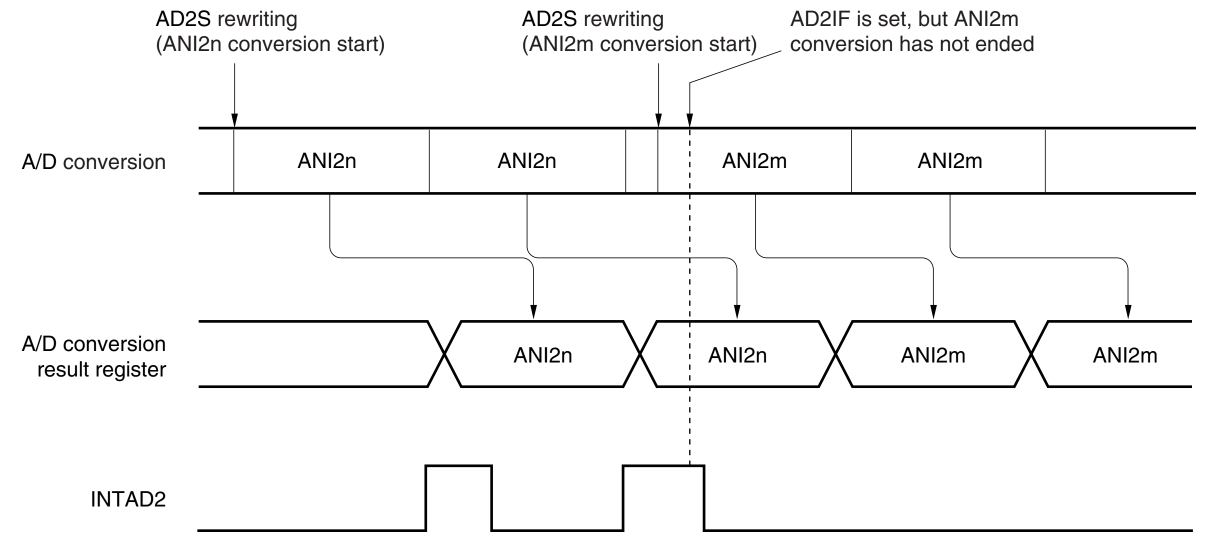
(4) Alternate input

The analog input pin (ANI2n) functions alternately as input port (P7n). When selecting one of the ANI2n pin to execute A/D conversion, do not execute an input instruction to port 7 during conversion as the conversion resolution may drop.

(5) Interrupt request flag (AD2IF)

The interrupt request flag (AD2IF) is not cleared even if the contents of the AD2S register are changed. If the analog input pin is changed during A/D conversion, therefore, the result of converting the previously selected analog input signal may be stored and the A/D2 conversion end interrupt request flag may be set immediately before the AD2S register is rewritten. If the AD2IF flag is read immediately after the AD2S register is rewritten, the AD2IF flag may be set even though the A/D conversion of the newly selected analog input pin has not been completed. When A/D conversion is stopped, clear the AD2IF flag before resuming conversion.

Figure 13-13. Generation Timing of A/D2 Conversion End Interrupt Request



Remark n= 0 to 11
m = 0 to 11

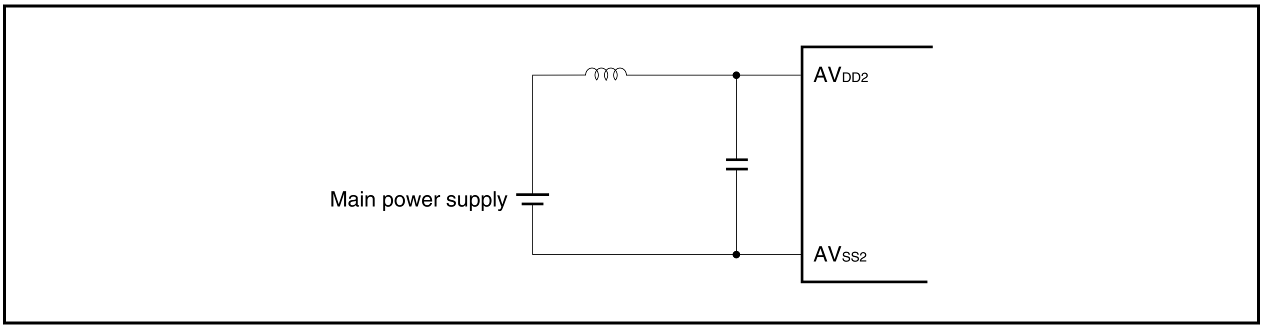
(6) AV$_{DD2}$ pin

(a) The AV$_{DD2}$ pin is used as the power supply pin of the A/D converter 2 and also supplies power to the alternate-function ports. In an application where a backup power supply is used, be sure to supply the same potential as EV$_{DD0}$, EV$_{DD1}$, EV$_{DD2}$, and EV$_{DD3}$ (V850E/IH4-H only) to the AV$_{DD2}$ pin as shown in Figure 13-12.

(b) The AV$_{DD2}$ pin is also used as the reference voltage pin of the A/D converter 2. If the source supplying power to the AV$_{DD2}$ pin has a high impedance or if the power supply has a low current supply capability, the reference voltage may fluctuate due to the current that flows during conversion (especially, immediately after the conversion operation enable (AD2CE bit = 1)). As a result, the conversion accuracy may drop. To avoid this, it is recommended to connect a capacitor across the AV$_{DD2}$ and AV$_{SS2}$ pins to suppress the reference voltage fluctuation as shown in Figure 13-14.

(c) If the source supplying power to the AV$_{DD2}$ pin has a high DC resistance (for example, because of insertion of a diode), the voltage when conversion is enabled may be lower than the voltage when conversion is stopped, because of a voltage drop caused by the A/D conversion current.

Figure 13-14. AV$_{DD2}$ Pin Connection Example



(7) Reading AD2CRn register

When the AD2M0, AD2M1, or AD2S register is written, the contents of the AD2CRn register may be undefined. Read the conversion result after completion of conversion and before writing to the AD2M0, AD2M1, and AD2S registers. The correct conversion result may not be read at a timing different from the above.

(8) A/D conversion result

If there is noise at the analog input pin (ANI2n) or at the power supply voltage pin (AV$_{DD2}$), that noise may generate an illegal conversion result.
Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.
An example of this software processing is shown below.

- Take the average result of a number of A/D conversions and use that as the A/D conversion result.
- Execute a number of A/D conversions successively and use those results, omitting any exceptional results that may have been obtained.
- If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing counteractive measures.

(9) Standby mode

Because the A/D converter 2 stops operating in the IDLE and STOP modes, conversion results are invalid, so power consumption can be reduced. Operations are resumed after the IDLE and STOP modes are released, but the A/D conversion results after the IDLE and STOP modes are released are invalid. When using the A/D converter 2 after the IDLE and STOP modes are released, before setting the IDLE and STOP modes or releasing the IDLE and STOP modes, set the AD2M0.AD2CE bit to 0 then set the AD2CE bit to 1 after releasing the IDLE and STOP modes.

(10) Variation of A/D conversion results

The results of the A/D conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. To reduce the variation, take counteractive measures with the program, such as by averaging the A/D conversion results.

(11) A/D conversion result hysteresis characteristics

Successive comparison type A/D converters hold an analog input voltage in an internal sample & hold capacitor and then perform A/D conversion. After the A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur if the output impedance from the analog input source is too high.

- When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value. Even if the conversion were to be performed at the same potential, the results may thus vary.
- When switching the analog input channel, hysteresis characteristics may appear where the conversion result is affected by the previous channel value. This is because one A/D converter is used for the A/D conversions. Even if the conversion were to be performed at the same potential, the results may thus vary.

To obtain more accurate conversion results, lower the output impedance from the analog input source or execute A/D conversion twice consecutively on the same channel, and discard the first conversion result.

13.8 How to Read A/D Converter Characteristics Table

For details about the A/D converter characteristics table, see **12.7 How to Read A/D Converter Characteristics Table**.

CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)

14.1 Features

- ○ Transfer rate: 300 bps to 1.25 Mbps (using peripheral clock (f_{XX}) of 100 MHz and dedicated baud rate generator)
- ○ Full-duplex communication: Internal UARTA receive data register n (UAnRX)
  Internal UARTA transmit data register n (UAnTX)
- ○ 2-pin configuration: TXDAn: Transmit data output pin
  RXDAn: Receive data input pin
- ○ Reception error output function
  - Parity error
  - Framing error
  - Overrun error
- ○ Interrupt sources: 3
  - Reception error interrupt (INTUAnRE): This interrupt is generated by ORing the three types of reception errors
  - Reception end interrupt (INTUAnR): This interrupt occurs upon transfer of receive data from the shift register to the UAnRX register after serial transfer end, in the reception enabled status.
  - Transmission enable interrupt (INTUAnT): This interrupt occurs upon transfer of transmit data from the UAnTX register to the shift register in the transmission enabled status.
- ○ Character length: 7, 8 bits
- ○ Parity function: Odd, even, 0, none
- ○ Transmission stop bit: 1, 2 bits
- ○ On-chip dedicated baud rate generator
- ○ MSB-/LSB-first transfer selectable
- ○ Transmit/receive data inverted input/output possible

Remark n = 0 to 2

14.2 Configuration

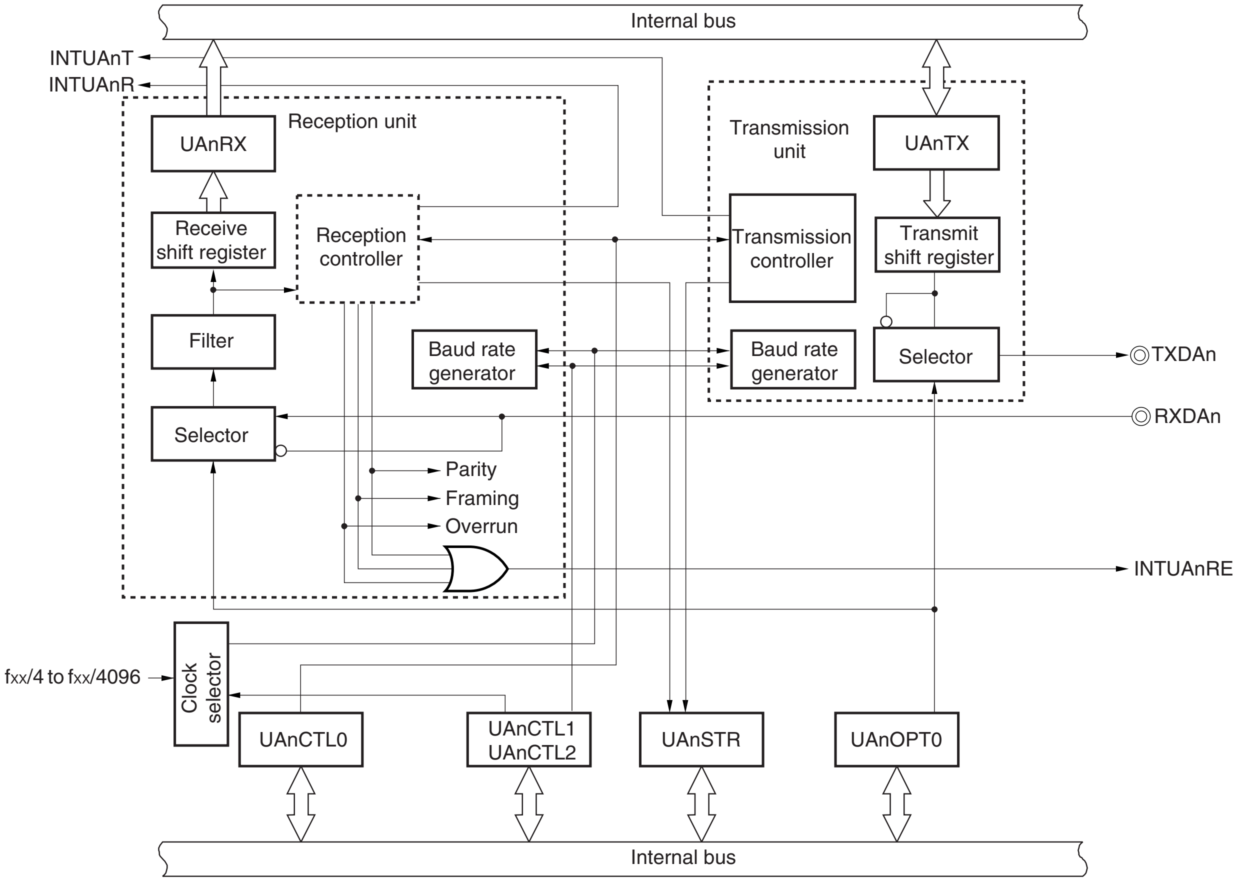
UARTAn consists of the following hardware units.

Table 14-1. Configuration of UARTAn

Item	Configuration
Registers	UARTAn control register 0 (UAnCTL0) UARTAn control register 1 (UAnCTL1) UARTAn control register 2 (UAnCTL2) UARTAn option control register 0 (UAnOPT0) UARTAn status register (UAnSTR) UARTAn receive shift register UARTAn receive data register (UAnRX) UARTAn transmit shift register UARTAn transmit data register (UAnTX)

The block diagram of the UARTAn is shown below.

Figure 14-1. Block Diagram of UARTAn



Remarks 1. n = 0 to 2

2. For the configuration of the baud rate generator, see **Figure 14-12**.

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register used to specify the UARTAn operation.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register used to select the base clock (f_{UCLK}) for the UARTAn.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register used to control the baud rate for the UARTAn.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register used to control serial transfer for the UARTAn.

(5) UARTAn status register (UAnSTR)

The UAnSTR register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (to 1) upon occurrence of a reception error.

(6) UARTAn receive shift register

This is a shift register used to convert the serial data input to the RXDAn pin into parallel data. Upon reception of 1 byte of data and detection of the stop bit, the receive data is transferred to the UAnRX register.
This register cannot be manipulated directly.

(7) UARTAn receive data register (UAnRX)

The UAnRX register is an 8-bit register that holds receive data. When 7 characters are received, 0 is stored in the highest bit (when data is received LSB first).
In the reception enabled status, receive data is transferred from the UARTAn receive shift register to the UAnRX register in synchronization with the completion of shift-in processing of 1 frame.
Transfer to the UAnRX register also causes the reception end interrupt request signal (INTUAnR) to be output.

(8) UARTAn transmit shift register

The UARTAn transmit shift register is a shift register used to convert the parallel data transferred from the UAnTX register into serial data.
When 1 byte of data is transferred from the UAnTX register, the UARTAn transmit shift register data is output from the TXDAn pin.
This register cannot be manipulated directly.

(9) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit transmit data buffer. Transmission starts when transmit data is written to the UAnTX register. When data can be written to the UAnTX register (when data of one frame is transferred from the UAnTX register to the UARTAn transmit shift register), the transmission enable interrupt request signal (INTUAnT) is generated.

14.2.1 Pin functions of each channel

The input and output pins used by UARTA in the V850E/IG4-H and V850E/IH4-H are alternately used for other functions as shown in Table 14-2. To use these pins for UARTA, set up the related registers as described in **Table 4-16 Settings When Pins Are Used for Alternate Functions**.

Table 14-2. Pins Used by UARTA

Channel	Pin No.		Port	UARTA Reception Input	UARTA Transmission Output	Other Functions
	IG4-H	IH4-H				
	GC	GF				
UARTA0	46	96	P40	RXDA0	–	SIF0/DDI/TOA00
	47	97	P41	–	TXDA0	SOF0
UARTA1	54	106	P30	RXDA1	–	SCL/$\overline{\text{WR1}}$
	55	107	P31	–	TXDA1	SDA/$\overline{\text{WAIT}}$
UARTA2	56	108	P32	RXDA2	–	SIF1/$\overline{\text{CS1}}$
	57	109	P33	–	TXDA2	SOF1

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

14.3 Mode Switching Between UARTA and Other Serial Interface

14.3.1 Mode switching between UARTA0 and CSIF0

In the V850E/IG4-H and V850E/IH4-H, UARTA0 and CSIF0 share a pin, and these functions cannot be used at the same time. To use the pin for the UARTA0 function, set up the PMC4, PFC4, and PFCE4 registers in advance.

Switching the operation mode between UARTA0 and CSIF0, the serial interfaces, is described below.

Caution The operations related to transmission and reception of UARTA0 or CSIF0 are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-2. Operation Mode Switch Settings of UARTA0 and CSIF0

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	0	0	PMC44	PMC43	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	PFC44	PFC43	0	PFC41	PFC40

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	0	0	0	0	0	PFCE42	0	PFCE40

PMC42	PFCE42	Operation mode
0	×	Port I/O mode
1	0	$\overline{\text{SCKF0}}$

PMC4n	PFC4n	Operation mode
0	×	Port I/O mode
1	0	CSIF0 mode
1	1	UARTA0 mode

Remarks 1. n = 0, 1

2. × = 0 or 1

14.3.2 Mode switching between UARTA1 and I^{2}C

In the V850E/IG4-H and V850E/IH4-H, UARTA1 and I^{2}C share a pin and these functions cannot be used at the same time. To use the pin for the UARTA1 function, set up the PMC3, PFC3, and PFCE3 registers in advance.

Switching the operation mode between UARTA1 and I^{2}C, the serial interfaces, is described below.

Caution The operations related to transmission and reception of UARTA1 or I^{2}C are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-3. Operation Mode Switch Settings of UARTA1 and I^{2}C

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	0	0	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC3n	PFC3n	Operation mode
0	×	Port I/O mode
1	0	UARTA1 mode
1	1	I^{2}C mode

Remarks 1. n = 0, 1

2. × = 0 or 1

14.3.3 Mode switching between UARTA2 and CSIF1

In the V850E/IG4-H and V850E/IH4-H, UARTA2 and CSIF1 share a pin, and these functions cannot be used at the same time. To use the pin for the UARTA2 function, set up the PMC3 and PFC3 registers in advance.

Switching the operation mode between UARTA2 and CSIF0, the serial interfaces, is described below.

Caution The operations related to transmission and reception of UARTA2 or CSIF1 are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-4. Operation Mode Switch Settings of UARTA2 and CSIF1

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	0	0	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC34	PFC34	Operation mode
0	×	Port I/O mode
1	0	$\overline{\text{SCKF1}}$ I/O

PMC3n	PFC3n	Operation mode
0	×	Port I/O mode
1	0	CSIF1 mode
1	1	UARTA2 mode

Remarks 1. n = 2, 3

2. × = 0 or 1

14.4 Control Registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that controls the UARTAn serial transfer operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 10H.

(1/2)

After reset: 10H R/W Address: UA0CTL0 FFFFFA00H, UA1CTL0 FFFFFA10H,
UA2CTL0 FFFFFA20H

	<7>	<6>	<5>	<4>	3	2	1	0
UAnCTL0 (n = 0 to 2)	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

UAnPWR	UARTAn operation control
0	Disable UARTAn operation (UARTAn reset asynchronously)
1	Enable UARTAn operation
The UARTAn operation is controlled by the UAnPWR bit. The TXDAn pin output is fixed to high level by clearing the UAnPWR bit to 0 (fixed to low level if UAnOPT0.UAnTDL bit = 1).	

UAnTXE	Transmission operation enable
0	Disable transmission operation
1	Enable transmission operation
• To start transmission, set the UAnPWR bit to 1 and then set the UAnTXE bit to 1. • To initialize the transmission unit, clear the UAnTXE bit to 0, wait for two cycles of the base clock (f_{UCLK}), and then set the UAnTXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **14.7 (1) (a) Base clock**). • When the operation is enabled (UAnPWR bit = 1), the transmission operation is enabled after two or more cycles of the base clock (f_{UCLK}) have elapsed since UAnTXE = 1. • When the UAnPWR bit is cleared to 0, the status of the internal circuit becomes the same status as UAnTXE bit = 0 by the UAnPWR bit even if the UAnTXE bit is 1. The transmission operation is enabled when the UAnPWR bit is set to 1 again.	

(2/2)

UAnRXE	Reception operation enable
0	Disable reception operation
1	Enable reception operation

- To start reception, set the UAnPWR bit to 1 and then set the UAnRXE bit to 1.
- To initialize the reception unit, clear the UAnRXE bit to 0, wait for two cycles of the base clock, and then set the UAnRXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **14.7 (1) (a) Base clock**).
- When the operation is enabled (UAnPWR bit = 1), the reception operation is enabled after two or more cycles of the base clock (f_{UCLK}) have elapsed since UAnRXE = 1. The start bit is ignored if it is received before the reception operation is enabled.
- When the UAnPWR bit is cleared to 0, the status of the internal circuit becomes the same status as UAnRXE bit = 0 by the UAnPWR bit even if the UAnRXE bit is 1. The reception operation is enabled when the UAnPWR bit is set to 1 again.

UAnDIR^Note	Transfer direction selection
0	MSB-first transfer
1	LSB-first transfer

UAnPS1^Note	UAnPS0^Note	Parity selection during transmission	Parity selection during reception
0	0	No parity output	Reception with no parity
0	1	0 parity output	Reception with 0 parity
1	0	Odd parity output	Odd parity check
1	1	Even parity output	Even parity check

If “reception with 0 parity” is selected during reception, a parity check is not performed. Therefore, since the UAnSTR.UAnPE bit is not set, no error interrupt due to a parity error is output.

UAnCL^Note	Specification of data character length of 1 frame of transmit/receive data
0	7 bits
1	8 bits

UAnSL^Note	Specification of length of stop bit for transmit data
0	1 bit
1	2 bits

Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnSL bit.

Note This register can be rewritten only when the UAnPWR bit = 0 or the UAnTXE bit = UAnRXE bit = 0. However, setting any or all of the UAnPWR, UAnTXE, and UAnRXE bits to 1 at the same time is possible.

Remark For details of parity, see **14.6.6 Parity types and operations**.

(2) UARTAn control register 1 (UAnCTL1)

For details, see **14.7 (2) UARTAn control register 1 (UAnCTL1)**.

(3) UARTAn control register 2 (UAnCTL2)

For details, see **14.7 (3) UARTAn control register 2 (UAnCTL2)**.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register that controls the serial transfer operation of UARTAn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 14H.

After reset: 14H R/W Address: UA0OPT0 FFFFFA03H, UA1OPT0 FFFFFA13H,
UA2OPT0 FFFFFA23H

	7	6	5	4	3	2	1	0
UAnOPT0 (n = 0 to 2)	0	0	0	1	0	1	UAnTDL	UAnRDL

UAnTDL	Transmit data level bit
0	Normal output of transfer data
1	Inverted output of transfer data

- The output level of the TXDAn pin can be inverted using the UAnTDL bit.
- This register can be set when the UAnCTL0.UAnPWR bit = 0 or when the UAnCTL0.UAnTXE bit = 0.

UAnRDL	Receive data level bit
0	Normal input of transfer data
1	Inverted input of transfer data

- The input level of the RXDAn pin can be inverted using the UAnRDL bit.
- This register can be set when the UAnPWR bit = 0 or the UAnCTL0.UAnRXE bit = 0.
- When the UAnRDL bit is set to 1 (inverted input of receive data), reception must be enabled (UAnCTL0.UAnRXE bit = 1) after setting the data reception pin to the UART reception pin (RXDAn) when reception is started. When the pin mode is changed after reception is enabled, the start bit will be mistakenly detected if the pin level is high.

Caution Be sure to set bits 3 and 5 to 7 to “0”, and set bits 2 and 4 to “1”. Operation with other settings is not guaranteed.

(5) UARTAn status register (UAnSTR)

The UAnSTR register is an 8-bit register that displays the UARTAn transfer status and reception error contents.

This register can be read or written in 8-bit or 1-bit units, but the UAnTSF bit is a read-only bit, while the UAnPE, UAnFE, and UAnOVE bits can both be read and written. However, these bits can only be cleared by writing 0; they cannot be set by writing 1 (even if 1 is written to them, the value is retained).

The initialization conditions are shown below.

Register/Bit	Initialization Conditions
UAnSTR register	• After reset • UAnCTL0.UAnPWR bit = 0
UAnTSF bit	• UAnCTL0.UAnTXE bit = 0
UAnPE, UAnFE, UAnOVE bits	• 0 write • UAnCTL0.UAnRXE bit = 0

Caution Be sure to read and check the error flags of the UAnPE, UAnFE, and UAnOVE bits, and clear the flags by writing “0” to them.

After reset: 00H R/W Address: UA0STR FFFFFA04H, UA1STR FFFFFA14H,
UA2STR FFFFFA24H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

(n = 0 to 2)

UAnTSF	Transfer status flag
0	• When the UAnPWR bit = 0 or the UAnTXE bit = 0 has been set. • When, following transfer end, there was no next data transfer from UAnTX register
1	Write to UAnTX register

The UAnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that the UAnTSF bit = 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while the UAnTSF bit = 1.

UAnPE	Parity error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set. • When 0 has been written
1	When parity of data and parity bit do not match during reception.

- The operation of the UAnPE bit is controlled by the settings of the UAnCTL0.UAnPS1 and UAnCTL0.UAnPS0 bits.
- The UAnPE bit can be read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

UAnFE	Framing error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set. • When 0 has been written
1	When no stop bit is detected during reception

- Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnCTL0.UAnSL bit.
- The UAnFE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

UAnOVE	Overrun error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set. • When 0 has been written
1	When receive data has been set to the UAnRX register and the next receive operation is ended before that receive data has been read.

- When an overrun error occurs, the data is discarded without the next receive data being written to the UAnRX register.
- The UAnOVE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

(6) UARTAn receive data register (UAnRX)

The UAnRX register is an 8-bit buffer register that stores parallel data converted by the UARTAn receive shift register.

The data stored in the UARTAn receive shift register is transferred to the UAnRX register upon end of reception of 1 byte of data. A reception end interrupt request signal (INTUAnR) is generated at this timing.

During LSB-first reception when the data length has been specified as 7 bits, the receive data is transferred to bits 6 to 0 of the UAnRX register and the MSB always becomes 0. During MSB-first reception, the receive data is transferred to bits 7 to 1 of the UAnRX register and the LSB always becomes 0.

When an overrun error occurs (UAnSTR.UAnOVE bit = 1), the receive data at this time is not transferred to the UAnRX register and is discarded.

This register is read-only in 8-bit units.

In addition to reset, the UAnRX register can be set to FFH by clearing the UAnCTL0.UAnPWR bit to 0.

After reset: FFH R Address: UA0RX FFFFFA06H, UA1RX FFFFFA16H,
UA2RX FFFFFA26H

	7	6	5	4	3	2	1	0
UAnRX (n = 0 to 2)								

(7) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit register used to set transmit data.

Transmission starts when transmit data is written to the UAnTX register in the transmission enabled status (UAnCTL0.UAnTXE bit = 1). Upon end of the transfer of the data of the UAnTX register to the UARTAn transmit shift register, the transmission enable interrupt request signal (INTUAnT) is generated.

This register can be read or written in 8-bit units.

Reset sets this register to FFH.

After reset: FFH R/W Address: UA0TX FFFFFA07H, UA1TX FFFFFA17H,
UA2TX FFFFFA27H

	7	6	5	4	3	2	1	0
UAnTX (n = 0 to 2)								

14.5 Interrupt Request Signals

The following three interrupt request signals are generated from UARTAn.

- Reception error interrupt request signal (INTUAnRE)
- Reception end interrupt request signal (INTUAnR)
- Transmission enable interrupt request signal (INTUAnT)

Among these three interrupt signals, the reception error interrupt signal has the highest default priority, and the reception end interrupt request signal and transmission enable interrupt request signal follow in this order.

Table 14-3. Interrupts and Their Default Priorities

Interrupt	Priority
Reception error	High
Reception end	⇳
Transmission enable	Low

(1) Reception error interrupt request signal (INTUAnRE)

A reception error interrupt request signal is generated while reception is enabled by ORing the three types of reception errors (parity error, framing error, and overrun error) explained in the UAnSTR register section.

(2) Reception end interrupt request signal (INTUAnR)

A reception end interrupt request signal is output when data is shifted into the UARTAn receive shift register and transferred to the UAnRX register in the reception enabled status.

No reception end interrupt request signal is generated in the reception disabled status.

(3) Transmission enable interrupt request signal (INTUAnT)

If transmit data is transferred from the UAnTX register to the UARTAn transmit shift register with transmission enabled, the transmission enable interrupt request signal is generated.

14.6 Operation

14.6.1 Data format

Full-duplex serial data reception and transmission is performed.

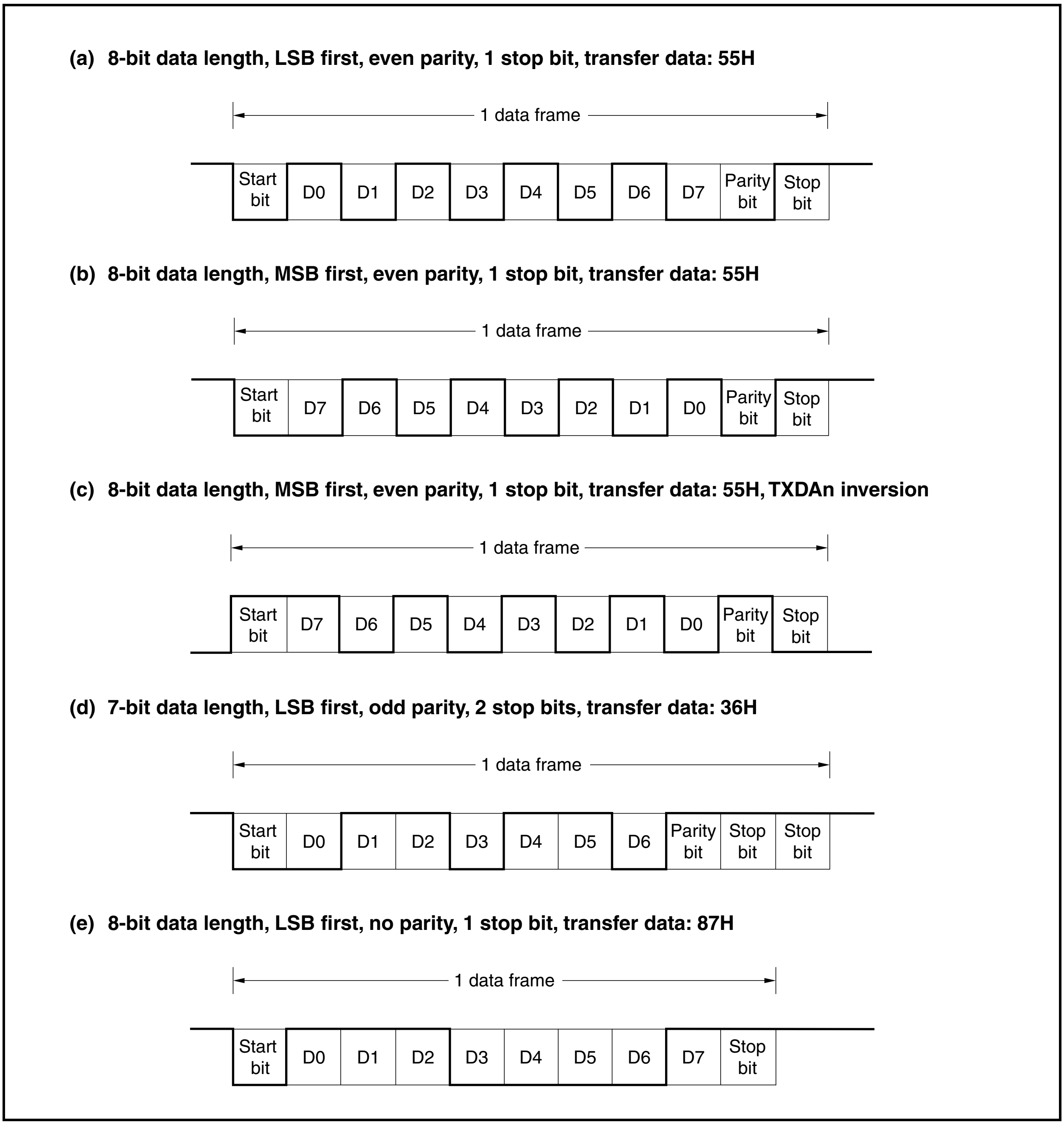
As shown in Figure 14-5, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB-/LSB-first transfer are performed using the UAnCTL0 register.

Moreover, control of UARTAn output/inverted output for the TXDAn pin is performed using the UAnOPT0.UAnTDL bit.

- Start bit..................1 bit
- Character bits........7 bits/8 bits
- Parity bit ....Even parity/odd parity/0 parity/no parity
- Stop bit ....1 bit/2 bits

Figure 14-5. UARTA Transmit/Receive Data Format



(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H

(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H

(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDAn inversion

(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H

(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H

14.6.2 UART transmission

A high level is output to the TXDAn pin by setting the UAnCTL0.UAnPWR bit to 1.

Next, the transmission enabled status is set by setting the UAnCTL0.UAnTXE bit to 1, and transmission is started by writing transmit data to the UAnTX register. The start bit, parity bit, and stop bit are automatically added.

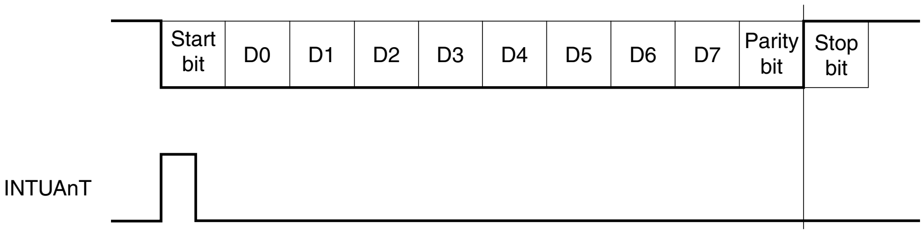
Since the CTS (transmit enable signal) input pin is not provided in UARTAn, use a port to check that reception is enabled at the transmit destination.

The data in the UAnTX register is transferred to the UARTAn transmit shift register upon the start of the transmit operation.

A transmission enable interrupt request signal (INTUAnT) is generated upon end of transmission of the data of the UAnTX register to the UARTAn transmit shift register, and thereafter the contents of the UARTAn transmit shift register are output to the TXDAn pin.

Write of the next transmit data to the UAnTX register is enabled by generating the INTUAnT signal.

Figure 14-6. UART Transmission



Remarks 1. LSB first

2. n = 0 to 2

14.6.3 Continuous transmission procedure

UARTAn can write the next transmit data to the UAnTX register when the UARTAn transmit shift register starts the shift operation. The transmit timing of the UARTAn transmit shift register can be judged from the transmission enable interrupt request signal (INTUAnT). An efficient communication rate is realized by writing the data to be transmitted next to the UAnTX register during transfer.

Caution During continuous transmission execution, perform initialization after checking that the UAnSTR.UAnTSF bit is 0. The transmit data cannot be guaranteed when initialization is performed while the UAnTSF bit is 1.

Remark n = 0 to 2

Figure 14-7. Continuous Transmission Processing Flow

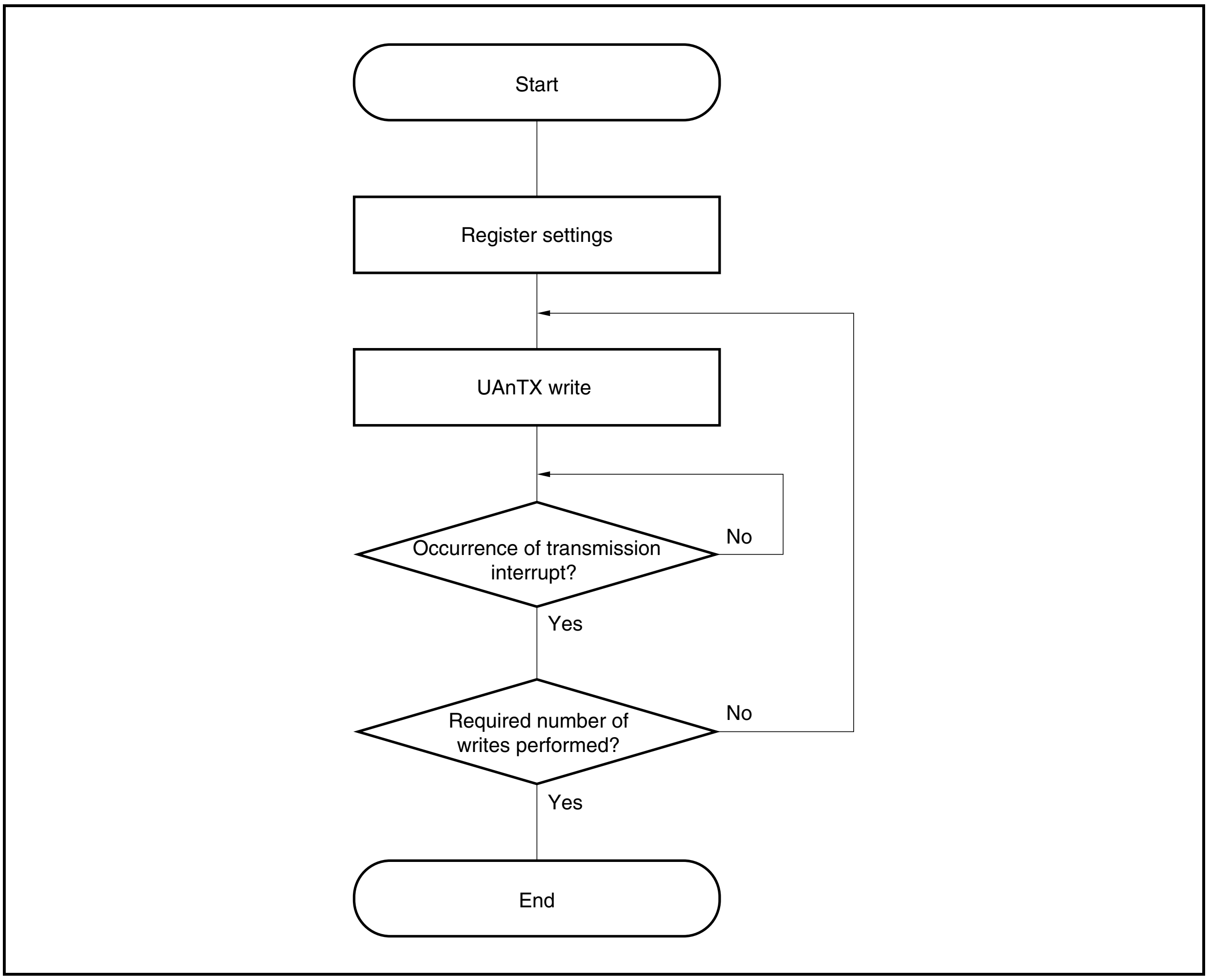
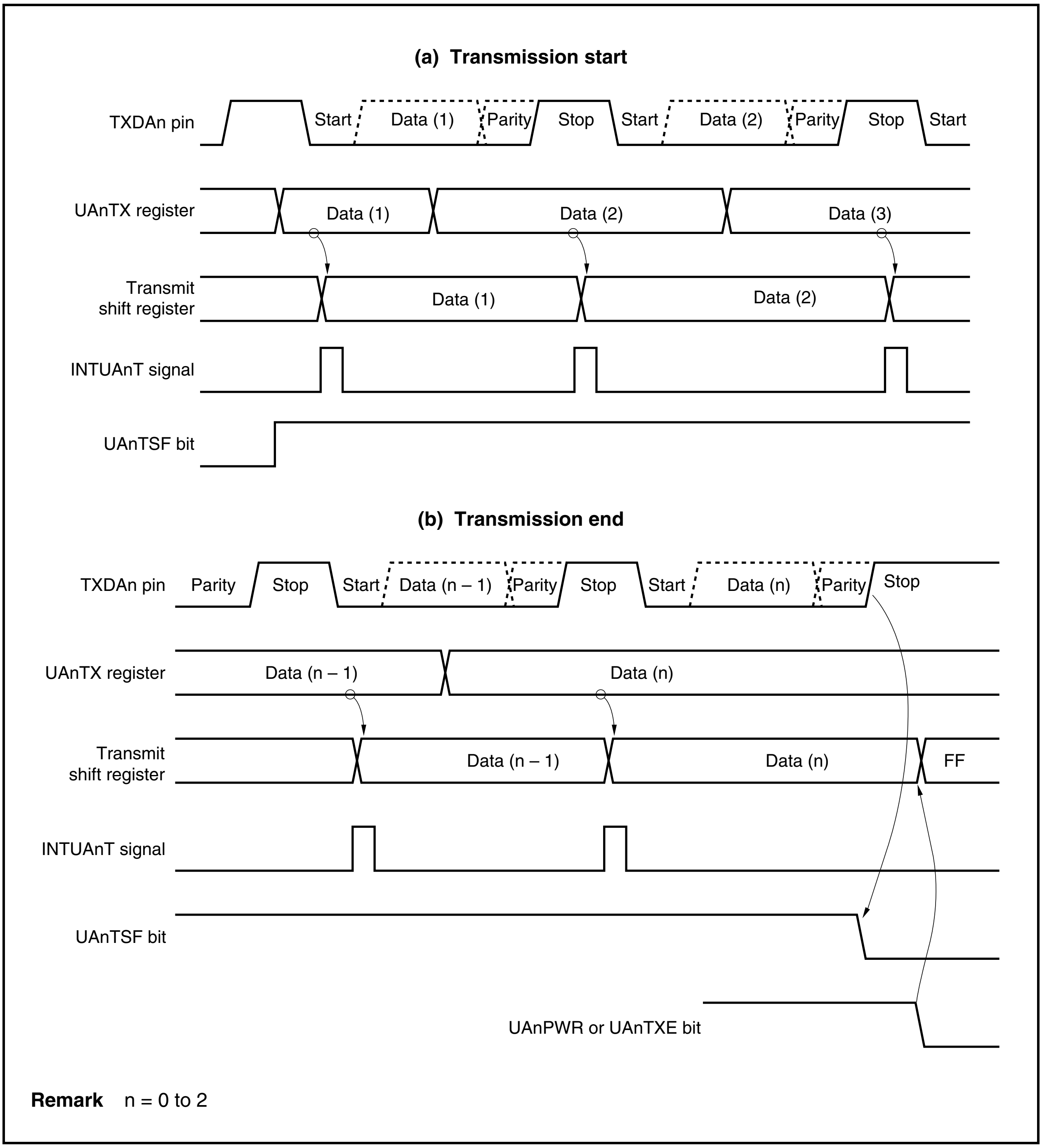

Start
Register settings
UAnTX write
Occurrence of transmission interrupt?
No
Yes
Required number of writes performed?
No
Yes
End

Figure 14-8. Continuous Transmission Operation Timing



14.6.4 UART reception

The reception wait status is set by setting the UAnCTL0.UAnPWR bit to 1 and then setting the UAnCTL0.UAnRXE bit to 1. In the reception wait status, the RXDAn pin is monitored and start bit detection is performed.

Start bit detection is performed using a two-step detection routine.

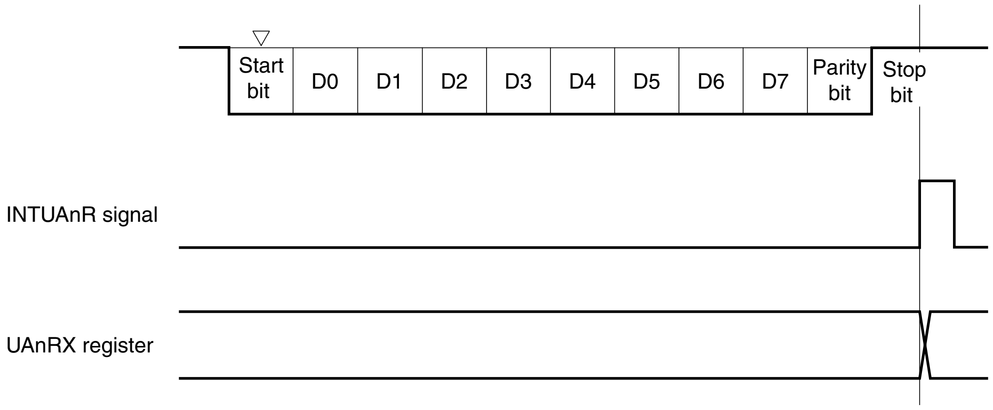
First the falling edge of the RXDAn pin is detected and sampling is started at the falling edge. The start bit is recognized if the RXDAn pin is low level at the start bit sampling point. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTAn receive shift register according to the set baud rate.

When the reception end interrupt request signal (INTUAnR) is output upon reception of the stop bit, the data of the UARTAn receive shift register is written to the UAnRX register. However, if an overrun error occurs (UAnSTR.UAnOVE bit = 1), the receive data at this time is not written to the UAnRX register and is discarded.

Even if a parity error (UAnSTR.UAnPE bit = 1) or a framing error (UAnSTR.UAnFE bit = 1) occurs during reception, reception continues until the reception position of the first stop bit, and the INTUAnRE signal is output following reception end.

Remark n = 0 to 2

Figure 14-9. UART Reception



Remark ▽: Start bit sampling point

Cautions 1. Be sure to read the UAnRX register even when a reception error occurs. If the UAnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.

2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

3. When reception is completed, read the UAnRX register after the reception end interrupt request signal (INTUAnR) has been generated, and clear the UAnPWR or UAnRXE bit to 0. If the UAnPWR or UAnRXE bit is cleared to 0 before the INTUAnR signal is generated, the read value of the UAnRX register cannot be guaranteed.

4. If receive end processing (INTUAnR signal generation) of UARTAn and the UAnPWR bit = 0 or UAnRXE bit = 0 conflict, the INTUAnR signal may be generated in spite of these being no data stored in the UAnRX register. To end reception without waiting INTUAnR signal generation, be sure to clear (0) the interrupt request flag (UAnRIC.UAnRIF), after setting (1) the interrupt mask flag (UAnRIC.UAnRMK) and then set (1) the UAnPWR bit = 0 or UAnRXE bit = 0.

14.6.5 Reception errors

Errors during a receive operation are of three types: parity errors, framing errors, and overrun errors. Data reception result error flags are set in the UAnSTR register and a reception error interrupt request signal (INTUAnRE) is output when an error occurs.

It is possible to ascertain which error occurred during reception by reading the contents of the UAnSTR register.

Clear the reception error flag by writing 0 to it after reading it.

Caution The reception end interrupt request signal (INTUAnR) and reception error interrupt request signal (INTUAnRE) are not generated simultaneously. The INTUAnR signal is generated when a reception ends normally. The INTUAnRE signal is generated and the INTUAnR signal is not generated when a reception error occurs.

Remark n = 0 to 2

- Reception error causes

Error Flag	Reception Error	Cause
UAnPE	Parity error	Received parity bit does not match the setting
UAnFE	Framing error	Stop bit not detected
UAnOVE	Overrun error	Reception of next data ended before data was read from UAnRX register

14.6.6 Parity types and operations

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(a) Even parity

(i) During transmission

The number of bits whose value is "1" among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.

- Odd number of bits whose value is "1" among transmit data: 1
- Even number of bits whose value is "1" among transmit data: 0

(ii) During reception

The number of bits whose value is "1" among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(b) Odd parity

(i) During transmission

Opposite to even parity, the number of bits whose value is "1" among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.

- Odd number of bits whose value is "1" among transmit data: 0
- Even number of bits whose value is "1" among transmit data: 1

(ii) During reception

The number of bits whose value is "1" among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(c) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(d) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

14.6.7 Receive data noise filter

This filter samples signals received via the RXDAn pin using the base clock (f_{UCLK}) supplied by the dedicated baud rate generator.

When the same sampling value is read twice, the match detector output changes and the RXDAn signal is sampled as the input data. Therefore, data not exceeding 1 clock cycle width is judged to be noise and is not delivered to the internal circuit (see **Figure 14-11**). See **14.7 (1) (a) Base clock** for details of the base clock.

Moreover, since the circuit is as shown in Figure 14-10, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Remark n = 0 to 2

Figure 14-10. Noise Filter Circuit

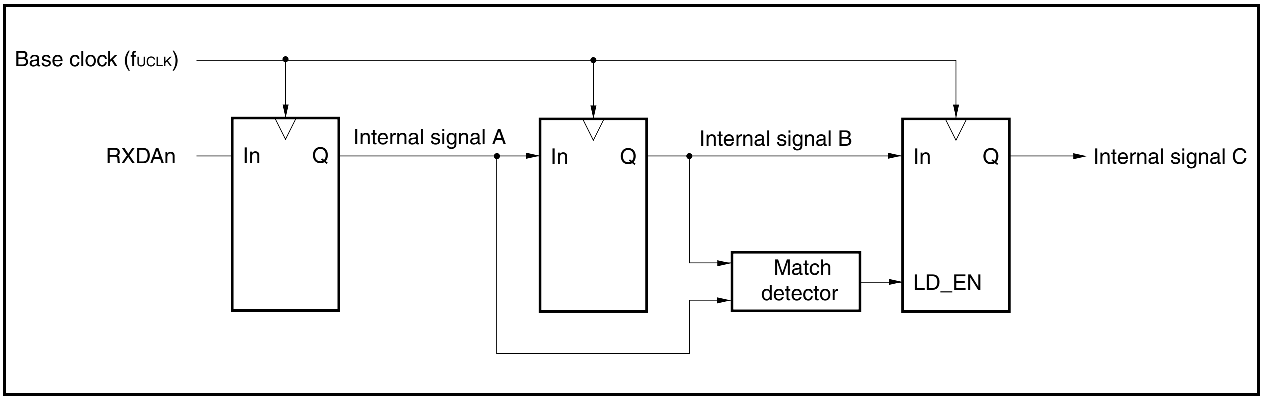
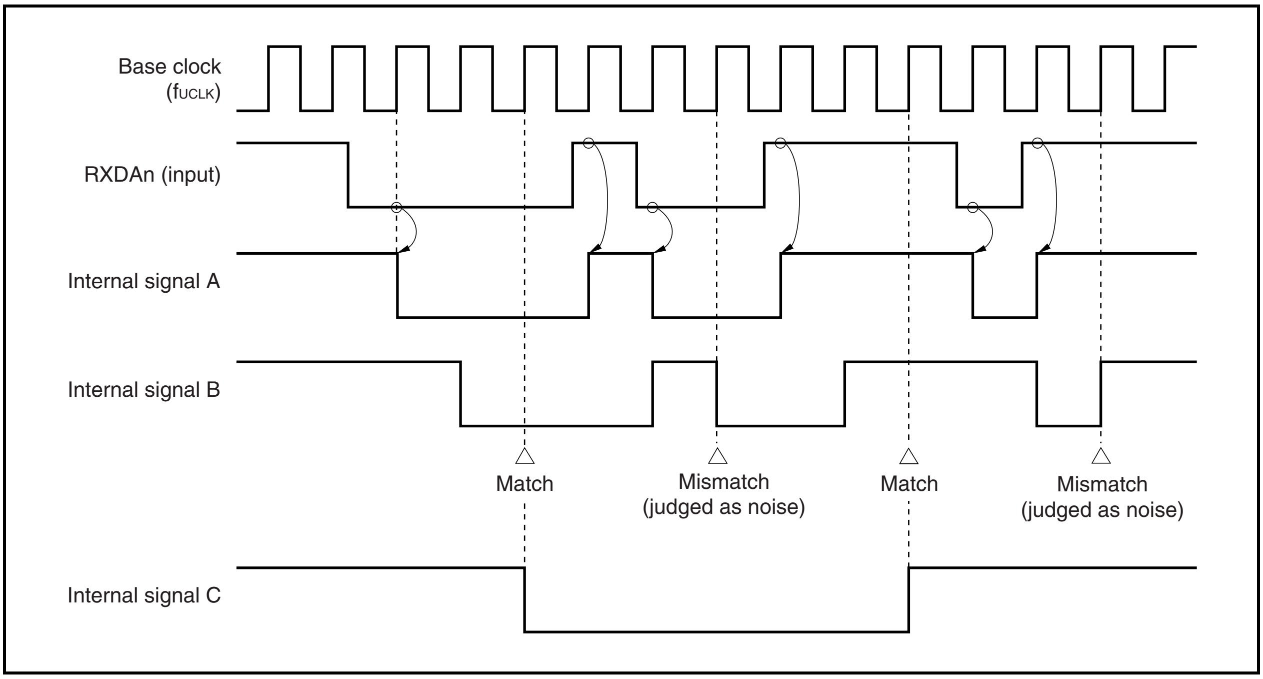


Figure 14-11. Timing of RXDAn Signal Judged as Noise



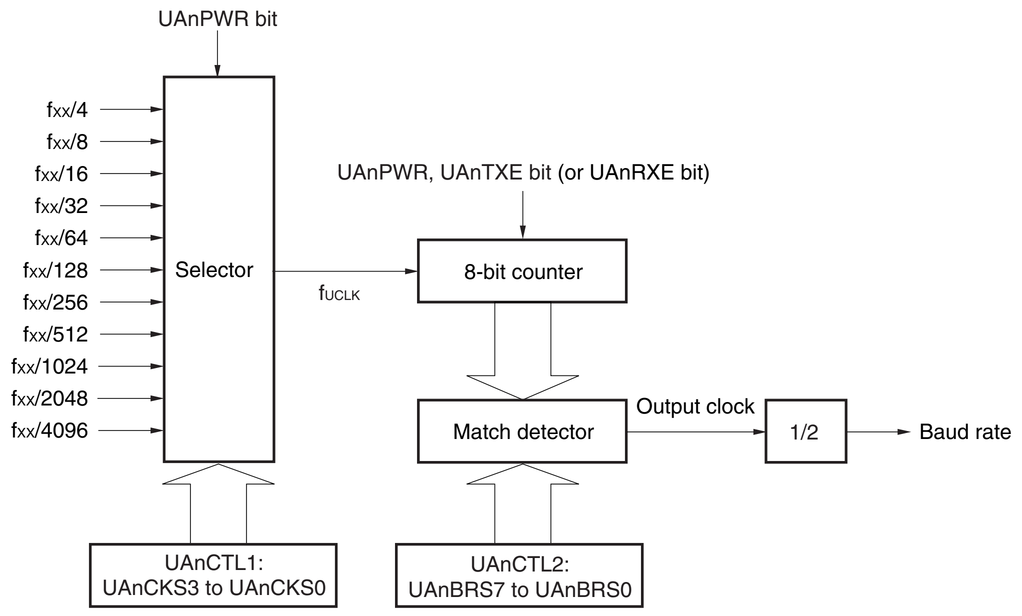
14.7 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a source clock selector block and an 8-bit programmable counter, and generates a serial clock during transmission and reception with UARTAn. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.

There is an 8-bit counter for transmission and another one for reception.

(1) Baud rate generator configuration

Figure 14-12. Configuration of Baud Rate Generator



Caution If the CPU clock (f_{CPU}) is slower than f_{UCLK}, UARTAn cannot be used.

Remarks 1. n = 0 to 2

2. fxx: Peripheral clock frequency

(a) Base clock

When the UAnCTL0.UAnPWR bit is 1, the clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits is supplied to the 8-bit counter. This clock is called the base clock (f_{UCLK}). When the UAnPWR bit = 0, f_{UCLK} is fixed to the low level.

(b) Serial clock generation

A serial clock can be generated by setting the UAnCTL1 register and the UAnCTL2 register.

The base clock (f_{UCLK}) is selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits.

The frequency division value for the 8-bit counter can be set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register that selects the UARTAn base clock.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution Clear the UAnCTL0.UAnPWR bit to 0 before rewriting the UAnCTL1 register.

After reset: 00H R/W Address: UA0CTL1 FFFFFA01H, UA1CTL1 FFFFFA11H,
UA2CTL1 FFFFFA21H

	7	6	5	4	3	2	1	0
UAnCTL1 (n = 0 to 2)	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	Base clock (f_{UCLK}) selection
0	0	0	0	$f_{xx}/4$
0	0	0	1	$f_{xx}/8$
0	0	1	0	$f_{xx}/16$
0	0	1	1	$f_{xx}/32$
0	1	0	0	$f_{xx}/64$
0	1	0	1	$f_{xx}/128$
0	1	1	0	$f_{xx}/256$
0	1	1	1	$f_{xx}/512$
1	0	0	0	$f_{xx}/1024$
1	0	0	1	$f_{xx}/2048$
1	0		0	$f_{xx}/4096$
Other than above				Setting prohibited

Remark f_{xx}: Peripheral clock frequency

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register that selects the baud rate (serial transfer speed) clock of UARTAn.

This register can be read or written in 8-bit units.

Reset sets this register to FFH.

Caution Clear the UAnCTL0.UAnPWR bit to 0 or clear the UAnTXE and UAnRXE bits to 00 before rewriting the UAnCTL2 register.

After reset: FFH R/W Address: UA0CTL2 FFFFFA02H, UA1CTL2 FFFFFA12H,
UA2CTL2 FFFFFA22H

	7	6	5	4	3	2	1	0
UAnCTL2 (n = 0 to 2)	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

UAn BRS7	UAn BRS6	UAn BRS5	UAn BRS4	UAn BRS3	UAn BRS2	UAn BRS1	UAn BRS0	Default (k)	Serial clock
0	0	0	0	0	0	×	×	–	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{UCLK}/4$
0	0	0	0	0	1	0	1	5	$f_{UCLK}/5$
0	0	0	0	0	1	1	0	6	$f_{UCLK}/6$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{UCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{UCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{UCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{UCLK}/255$

Remark f_{UCLK}: Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits

(4) Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{UCLK}}{2 \times k} \text{ [bps]}$$

f_{UCLK}: Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits

k: Value set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (k = 4, 5, 6, ..., 255)

(5) Baud rate error

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. The baud rate error during transmission must be within the error tolerance on the receiving side.

2. The baud rate error during reception must satisfy the range indicated in section (7) Allowable baud rate range during reception.

Example Peripheral clock frequency = 100 MHz = 100,000,000 Hz
Set value of UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits = 0000B (f_{UCLK} = 25,000,000 Hz)
Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits = 01010001B (k = 81)
Target baud rate = 153,600

Baud rate = 25,000,000/ (2 × 81)
= 154,321 [bps]

Error = (154,321/153,600 − 1) × 100
= 0.47 [%]

(6) Baud rate setting example

Table 14-4. Baud Rate Generator Setting Data

Baud Rate (bps)	f_{XX} = 100 MHz		
	UAnCTL1	UAnCTL2	ERR (%)
300	08H	A2H	0.47
600	07H	A2H	0.47
1,200	06H	A2H	0.47
2,400	05H	A2H	0.47
4,800	04H	A2H	0.47
9,600	03H	A2H	0.47
19,200	02H	A2H	0.47
31,250	02H	64H	0
38,400	01H	A2H	0.47
76,800	00H	A2H	0.47
153,600	00H	51H	0.47
312,500	00H	28H	0.00
625,000	00H	14H	0.00
1,250,000	00H	0AH	0.00

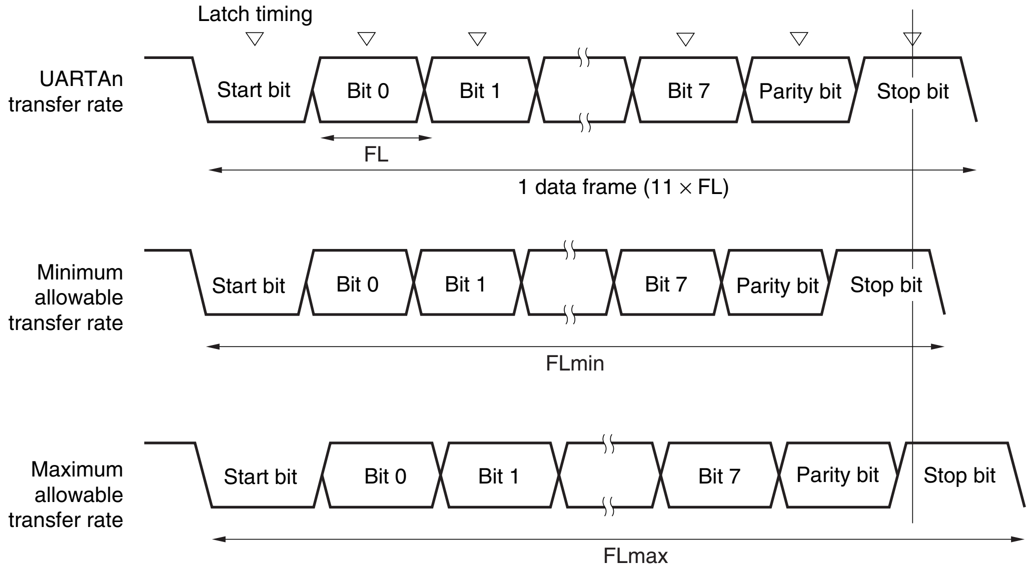
Remark f_{XX}: Peripheral clock frequency
ERR: Baud rate error (%)

(7) Allowable baud rate range during reception

The baud rate error range at the destination that is allowable during reception is shown below.

Caution The baud rate error during reception must be set within the allowable error range using the following equation.

Figure 14-13. Allowable Baud Rate Range During Reception



Remark n = 0 to 2

As shown in Figure 14-13, the receive data latch timing is determined by the counter set using the UAnCTL2 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception, the following is the theoretical result.

$FL = (Brate)^{-1}$

Brate: UARTAn baud rate (n = 0 to 2)

k: Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 2)

FL: 1-bit data length

Latch timing margin: 2 clocks

Minimum allowable transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k}\ FL$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$\mathrm{BRmax} = (\mathrm{FLmin}/11)^{-1} = \frac{22k}{21k + 2}\ \mathrm{Brate}$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times \mathrm{FLmax} = 11 \times \mathrm{FL} - \frac{k + 2}{2 \times k} \times \mathrm{FL} = \frac{21k - 2}{2 \times k}\ \mathrm{FL}$$

$$\mathrm{FLmax} = \frac{21k - 2}{20\ k}\ \mathrm{FL} \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$\mathrm{BRmin} = (\mathrm{FLmax}/11)^{-1} = \frac{20k}{21k - 2}\ \mathrm{Brate}$$

Obtaining the allowable baud rate error for UARTAn and the destination from the above-described equations for obtaining the minimum and maximum baud rate values yields the following.

Table 14-5. Maximum/Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.32%	−2.43%
8	+3.52%	−3.61%
20	+4.26%	−4.30%
50	+4.56%	−4.58%
100	+4.66%	−4.67%
255	+4.72%	−4.72%

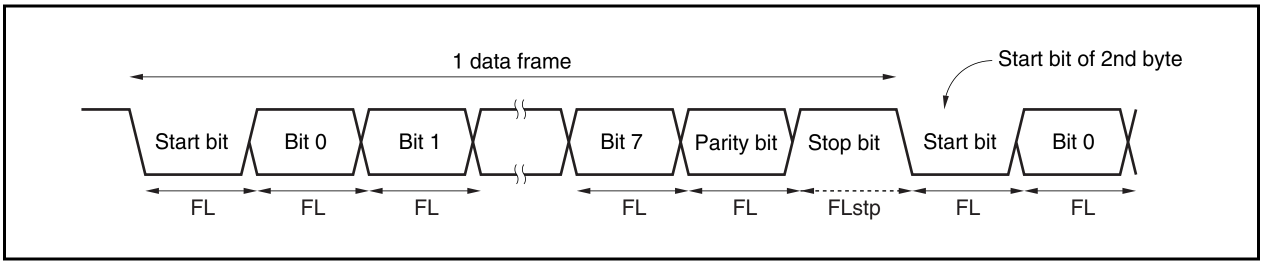
Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. k: Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 2)

(8) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from the stop bit to the next start bit is usually 2 base clocks longer. However, timing initialization is performed via start bit detection by the receiving side, so this has no influence on the transfer result.

Figure 14-14. Transfer Rate During Continuous Transmission



Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency: f_{UCLK}, we obtain the following equation.

$$FLstp = FL + 2/f_{UCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + (2/f_{UCLK})$$

14.8 Cautions

When the clock supply to UARTAn is stopped (for example, in IDLE or STOP mode), the operation stops with each register retaining the value it had immediately before the clock supply was stopped. The TXDAn pin output also holds and outputs the value it had immediately before the clock supply was stopped. However, the operation is not guaranteed after the clock supply is resumed. Therefore, after the clock supply is resumed, the circuits should be initialized by setting the UAnCTL0.UAnPWR, UAnCTL0.UAnRXE, and UAnCTL0.UAnTXE bits to 000.

Remark n = 0 to 2

CHAPTER 15 ASYNCHRONOUS SERIAL INTERFACE B (UARTB)

15.1 Features

- Transfer rate: Maximum 5 Mbps (using a dedicated baud rate generator)
- Full-duplex communications
- Single mode and FIFO mode selectable
  - Single mode: 8-bit × 1-stage data register (UBTX register or UBRX register) is used for each of transmission and reception.
  - FIFO mode
    Transmit FIFO: UBTX register (8 bits × 16 stages).
    Receive FIFO: UBRXAP register (16 bits × 16 stages)
    2 bits of the higher 8 bits of the UBRXAP register are for an error flag.
- Two-pin configuration
  TXDB: Transmit data output pin
  RXDB: Receive data input pin
- Reception error detection function
  - Overflow error (FIFO mode only)
  - Parity error
  - Framing error
  - Overrun error (single mode only)
- Interrupt sources: 5 types
  - Reception error interrupt request signal (INTUBTIRE)
  - Reception end interrupt request signal (INTUBTIR)
  - Transmission enable interrupt request signal (INTUBTIT)
  - FIFO transmission end interrupt request signal (INTUBTIF) (FIFO mode only)
  - Reception timeout interrupt request signal (INTUBTITO) (FIFO mode only)
- The character length of transmit/receive data is specified according to the UBCTL0 register
- Character length: 7 or 8 bits
- Parity functions: Odd, even, 0, or none
- Transmission stop bits: 1 or 2 bits
- MSB first/LSB first selectable for transfer data
- On-chip dedicated baud rate generator

15.2 Configuration

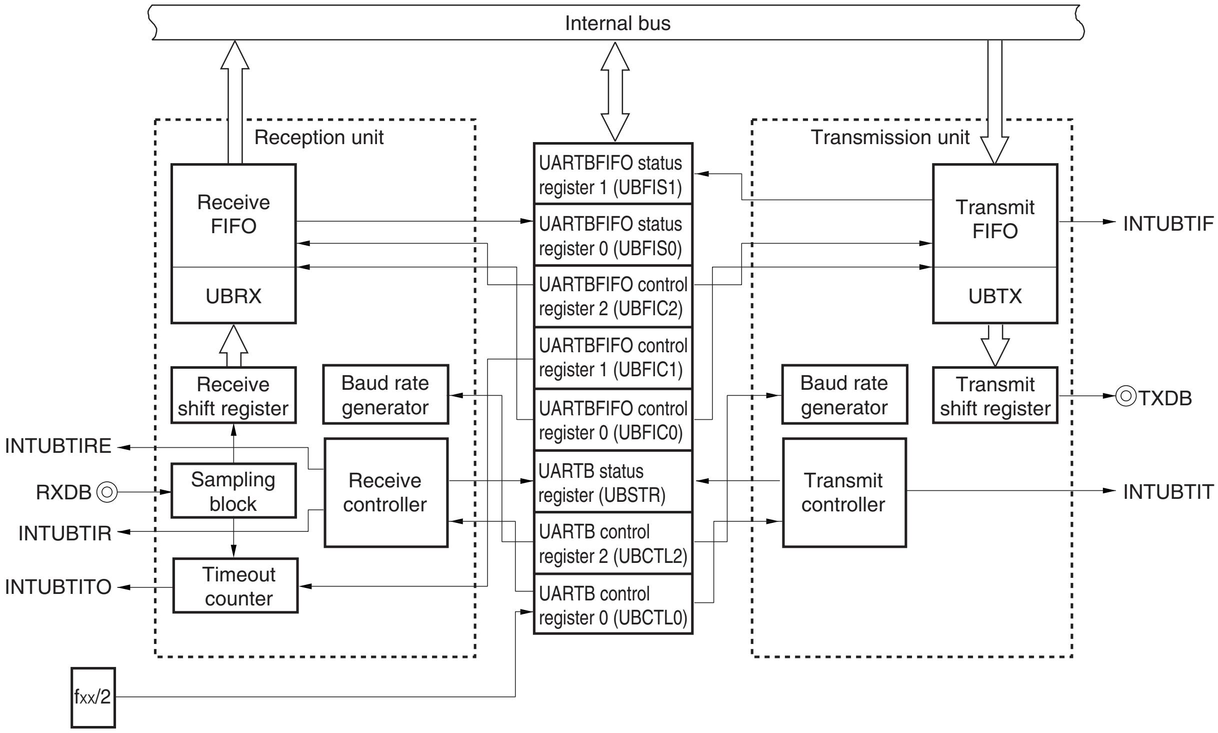
UARTB consists of the following hardware units.

Table 15-1. Configuration of UARTB

Item	Configuration
Registers	UARTB control register 0 (UBCTL0) UARTB control register 2 (UBCTL2) UARTB status register (UBSTR) UARTB FIFO control register 0 (UBFIC0) UARTB FIFO control register 1 (UBFIC1) UARTB FIFO control register 2 (UBFIC2) UARTB FIFO status register 0 (UBFIS0) UARTB FIFO status register 1 (UBFIS1) Receive shift register UARTB receive data register AP (UBRXAP) UARTB receive data register (UBRX) Transmit shift register UARTB transmit data register (UBTX)

The block diagram of the UARTB is shown below.

Figure 15-1. Block Diagram of UARTB



Remarks 1. fxx: Peripheral clock

2. For the configuration of the baud rate generator, see **Figure 15-9**.

(1) UARTB control register 0 (UBCTL0)

This register controls the transfer operation of UARTB.

(2) UARTB status register (UBSTR)

This register indicates the transfer status during transmission and the contents of a reception error. The status flag of this register, which indicates the transfer status during transmission, indicates the data retention status of the transmit shift register and the transmit data register (the UBTX register in the single mode or transmit FIFO in the FIFO mode). Each reception error flag is set to 1 when a reception error occurs, and cleared to 0 when 0 is written to the UBSTR register.

(3) UARTB control register 2 (UBCTL2)

This register is used to specify the division ratio by which to control the baud rate (serial transfer speed) of UARTB.

(4) UARTB FIFO control register 0 (UBFIC0)

This register is used to select the operation mode of UARTB, clear the transmit FIFO/receive FIFO that becomes valid in the FIFO mode, and specify the timing mode in which the transmission enable interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR) occurs.

(5) UARTB FIFO control register 1 (UBFIC1)

This register is valid in the FIFO mode. It generates a reception timeout interrupt request signal (INTUBTITO) if data is stored in the receive FIFO when the next data does not come (start bit is not detected) even after the reception wait time of the next data has elapsed after the stop bit has been received.

(6) UARTB FIFO control register 2 (UBFIC2)

This register is valid in the FIFO mode. It is used to set the timing to generate the transmission enable interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR), using the number of data transmitted or received as a trigger.

(7) UARTB FIFO status register 0 (UBFIS0)

This register is valid in the FIFO mode. The number of bytes of data stored in the receive FIFO can be read from this register.

(8) UARTB FIFO status register 1 (UBFIS1)

This register is valid in the FIFO mode. The number of empty bytes of the transmit FIFO can be read from this register.

(9) Receive shift register

This is a shift register that converts the serial data that was input to the RXDB pin into parallel data. One byte of data is received, and if a stop bit is detected, the received data is transferred to the receive data register.

This register cannot be directly manipulated.

(10) UARTB receive data register AP (UBRXAP), UARTB receive data register (UBRX)

The receive data register holds receive data. In the single mode, the 8-bit × 1-stage UBRX register is used. The 16-bit × 16-stage receive FIFO (UBRXAP register) is used in the FIFO mode.

The receive data is stored in the lower 8 bits of the receive FIFO (UBRXAP register) and the error information of the received data is stored in the higher 8 bits (bit 8 and bit 9). If a reception error (such as a parity error or a framing error) occurs in the FIFO mode, the error data can be identified by reading the UBRXAP register in 16-bit (halfword) units (error information is appended as UBPEF bit = 1 or UBFEF bit = 1). When the lower 8 bits of the UBRXAP register are read in 8-bit (byte) units, the higher 8 bits are discarded. Therefore, if no error has occurred, only the receive data of the UBRXAP register can be read successively by being read in 8-bit (byte) units in the same way as the UBRX register.

When 7-bit length data is received with the LSB first, the received data is transferred to bits 6 to 0 of the receive data register from the LSB (bit 0), with the MSB (bit 7) always being 0. When data is received with the MSB first, the received data is transferred to bits 7 to 1 of the receive data register from the MSB (bit 7), with the LSB (bit 0) always being 0. If an overrun error occurs, the receive data at that time is not transferred to the receive data register.

While reception is enabled, the received data is transferred from the receive shift register to the receive data register, in synchronization with the shift-in processing of one frame.

A reception end interrupt request signal (INTUBTIR) is generated by transferring the data to the UBRX register in the single mode, or transferring the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to receive FIFO in the FIFO mode. If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

(11) Transmit shift register

This is a shift register that converts the parallel data that was transferred from the transmit data register into serial data.

When one byte of data is transferred from the transmit data register, the transmit shift register data is output from the TXDB pin.

This register cannot be directly manipulated.

(12) UARTB transmit data register (UBTX)

The transmit data register is a buffer for transmit data. The 8-bit × 1-stage UBTX register is used as this buffer in the single mode. In the FIFO mode, the 8-bit × 16-stage transmit FIFO is used.

When 7-bit length data is transmitted with the LSB first, bits 6 to 0 of the transmit data register are transmitted as the transmit data from the LSB (bit 0) with the MSB (bit 7) always being 0. When data is transmitted with the MSB first, bits 7 to 1 of the transmit data register are transmitted as the transmit data from the MSB (bit 7) with the LSB (bit 0) always being 0.

In the single mode, transmission is started by writing transmit data to the UBTX register while transmission is enabled (UBCTL0.UBTXE bit = 1). When writing the transmit data to the UBTX register is enabled (when 1-byte data is transferred from the UBTX register to the transmit shift register), a transmission enable interrupt request signal (INTUBTIT) is generated.

In the FIFO mode, transmission is started by writing at least the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less to transmit FIFO and then enabling transmission (UBTXE bit = 1). When the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register (transmit data of the number set as the trigger can be written), a transmission enable interrupt request signal (INTUBTIT) is generated. In the FIFO mode, a FIFO transmission enable interrupt request signal (INTUBTIF) is generated when there is no more data in transmit FIFO and the transmit shift register (when FIFO and the register become empty).

(13) Timeout counter

This counter is used to recognize that data exists (remains) in receive FIFO when the number of received data does not reach the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits, and is valid only in the FIFO mode.

If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed after the stop bit has been received, a reception timeout interrupt request signal (INTUBTITO) is generated.

(14) Sampling block

This block samples the RXDB signal at the rising edge of the input clock ($f_{XX}/2$). If the same sampling value is detected two times, output of the match detector changes, and the value is sampled as input data. Data of less than one clock width is judged as noise and is not transmitted to the internal circuitry.

15.2.1 Pin functions of each channel

The RXDB and TXDB pins used by UARTB in the V850E/IG4-H and V850E/IH4-H are used alternately for other functions as shown in Table 15-2. To use these pins for UARTB, set up the related registers as described in **Table 4-16 Settings When Pins Are Used for Alternate Functions**.

Table 15-2. Pins Used by UARTB

Pin Number		Port	UARTB Reception Input	UARTB Transmission Output	Other Alternate Function
IG4-H	IH4-H				
GC	GF				
59	111	P35	RXDB	–	SIF2
60	112	P36	–	TXDB	SOF2

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H) : 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H) : 128-pin plastic LQFP (fine pitch) (14 × 20)

15.3 Mode Switching Between UARTB and CSIF2

In the V850E/IG4-H and V850E/IH4-H, UARTB, CSIF2, and PFCE3 share a pin, and these functions cannot be used at the same time. When using UARTB, set up the PMC3, PFC3, and PFCE3 registers in advance.

Switching the operation mode between UARTB and CSIF2, the serial interfaces, is described below.

Caution The operations related to transmission and reception of UARTB or CSIF2 are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 15-2. Operation Mode Switch Settings of UARTB and CSIF2

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	0	0	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC35	PFC35	Operation mode
0	×	Port I/O mode
1	0	CSIF2 mode
1	1	UARTB mode

Remark x = 0 or 1

15.4 Control Registers

(1) UARTB control register 0 (UBCTL0)

The UBCTL0 register controls the transfer operations of UARTB.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 10H.

Cautions 1. When using UARTB, set the external pins related to the UARTB function in the alternate-function mode, set UARTB control register 2 (UBCTL2). Then set the UBPWR bit to 1 before setting the other bits.

2. Be sure to input a high level to the RXDB pin when setting the external pins related to the UARTB function in the alternate-function mode. If a low level is input, it is judged that a falling edge is input after the UBRXE bit has been set to 1, and reception may be started.

Remark When reception is disabled, the receive shift register does not detect a start bit. No shift-in processing or transfer processing to the receive data register is performed, and the contents of the receive data register are retained.

When reception is enabled, the receive shift operation starts, in synchronization with the detection of the start bit, and when the reception of one frame is completed, the contents of the receive shift register are transferred to the receive data register.

A reception end interrupt request signal (INTUBTIR) is also generated, in synchronization with the transfer to the receive data register (in FIFO mode, transfer triggered by reaching set number of receive data).

If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

(1/2)

After reset: 10H R/W Address: FFFFFA40H

	<7>	<6>	<5>	<4>	3	2	1	0
UBCTL0	UBPWR	UBTXE	UBRXE	UBDIR	UBPS1	UBPS0	UBCL	UBSL

UBPWR	Operation clock control to UARTB
0	Stops supply of clocks to UARTB
1	Supplies clocks to UARTB

- When the UBPWR bit is cleared to 0, the UARTB can be asynchronously reset.
- When the UBPWR bit = 0, UARTB is in a reset state. Therefore, to operate UARTB, the UBPWR bit must be set to 1.
- When the UBPWR bit is changed from 1 to 0, all registers of UARTB are initialized. When the UBPWR bit is set to 1 again, the UARTB registers must be set again.
- The TXDB pin output is high level when the UBPWR bit is cleared to 0.

UBTXE	Transmission enable
0	Transmission is disabled
1	Transmission is enabled

- On startup, set the UBPWR bit to 1 and then set the UBTXE bit to 1. To stop transmission, clear the UBTXE bit to 0 and then the UBPWR bit to 0.
- When the transmission unit status is to be initialized, the transmission status may not be able to be initialized unless the UBTXE bit is set to 1 again after an interval of two cycles of $f_{xx}/2$ has elapsed since the UBTXE bit was cleared to 0.

UBRXE	Reception enable
0	Reception is disabled
1	Reception is enabled

- On startup, set the UBPWR bit to 1 and then set the UBRXE bit to 1. To stop reception, clear the UBRXE bit to 0 and then the UBPWR bit to 0.
- When the reception unit status is to be initialized, the reception status may not be able to be initialized unless the UBRXE bit is set to 1 again after an interval of two cycles of $f_{xx}/2$ has elapsed since the UBRXE bit was cleared to 0.

(2/2)

UBDIR	Specification of transfer direction mode (MSB/LSB)
0	MSB transfer first
1	LSB transfer first

- Clear the UBPWR bit or UBTXE and UBRXE bits to 0 before changing the setting of the UBDIR bit.

UBPS1	UBPS0	Parity selection during transmission	Parity selection during reception
0	0	Do not output a parity bit	Receive with no parity
0	1	Output 0 parity	Receive as 0 parity
1	0	Output odd parity	Judge as odd parity
1	1	Output even parity	Judge as even parity

- Clear the UBTXE and UBRXE bits to 0 before overwriting the UBPS1 and UBPS0 bits.
- If “0 parity” is selected for reception, no parity judgment is made. Therefore, no error interrupt is generated because the UBSTR.UBPE bit is not set to 1.

UBCL	Specification of data character length of 1-frame transmit/receive data
0	7 bits
1	8 bits

Clear the UBTXE and UBRXE bits to 0 before overwriting the UBCL bit.

UBSL	Specification of stop bit length of transmit data
0	1 bit
1	2 bits

- Clear the UBTXE bit to 0 before overwriting the UBSL bit.
- Since reception always operates by using a single stop bit length, the UBSL bit setting does not affect receive operations.

Remark For details of parity, see **15.7.6 Parity types and corresponding operation**.

(2) UARTB status register (UBSTR)

The UBSTR register indicates the transfer status and reception error contents while UARTB is transmitting data.

The status flag that indicates the transfer status during transmission indicates the data retention status of the transmit shift register and transmit data register (the UBTX register in the single mode or transmit FIFO in the FIFO mode). The status flag that indicates a reception error holds its status until it is cleared to 0.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution When the UBCTL0.UBPWR bit or UBCTL0.UBRXE bit is set to 0, or when 0 is written to the UBSTR register, the UBSTR.UBOVF, UBSTR.UBPE, UBSTR.UBFE, and UBSTR.UBOVE bits are cleared to 0.

(1/2)

After reset: 00H R/W Address: FFFFFA44H

	<7>	6	5	4	3	<2>	<1>	<0>
UBSTR	UBTSF	0	0	0	UBOVF	UBPE	UBFE	UBOVE

UBTSF	Transfer status flag
0	• In single mode (UBFIC0.UBMOD bit = 0) Data to be transferred to the transmit shift register and UBTX register does not exist (cleared (0) when UBCTL0.UBPWR bit = 0 or UBCTL0.UBTXE bit = 0). • In FIFO mode (UBFIC0.UBMOD bit = 1) Data to be transferred to the transmit shift register and transmit FIFO does not exist (cleared (0) when UBCTL0.UBPWR bit = 0 or UBCTL0.UBTXE bit = 0).
1	• In single mode (UBFIC0.UBMOD bit = 0) Data to be transferred to the transmit shift register or UBTX register exists (transmission in progress). • In FIFO mode (UBFIC0.UBMOD bit = 1) Data to be transferred to the transmit shift register and transmit FIFO exists (transmission in progress).

The value of the UBTSF bit is reflected after two periods of $f_{XX}/2$ have elapsed, after the transmit data is written to the UBTX register. Therefore, exercise care when referencing the UBTSF bit after transmit data has been written to the UBTX register.

UBOVF	Overflow flag
0	Overflow did not occur.
1	Overflow occurred (during reception).

- The UBOVF bit is valid only in the FIFO mode (when UBFIC0.UBMOD bit = 1), and invalid in the single mode (when UBFIC0.UBMOD bit = 0).
- If an overflow occurs, the received data is not written to receive FIFO but discarded.

(2/2)

UBPE	Parity error flag
0	Parity error did not occur.
1	Parity error occurred (during reception).

- The UBPE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- The operation of the UBPE bit differs according to the settings of the UBCTL0.UBPS1 and UBCTL0.UBPS0 bits.

UBFE	Framing error flag
0	Framing error did not occur.
1	Framing error occurred (during reception).

- The UBFE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- Only the first bit of the stop bits of the receive data is checked, regardless of the stop bit length.

UBOVE	Overrun error flag
0	Overrun error did not occur.
1	Overrun error occurred (during reception).

- The UBOVE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- When an overrun error occurs, the next receive data value is not written to the UBRX register and the data is discarded.

(3) UARTB control register 2 (UBCTL2)

The UBCTL2 register is used to specify the division ratio by which to control the baud rate (serial transfer speed) of UARTB.

This register can be read or written in 16-bit units.

Reset sets this register to FFFFH.

Caution When rewriting the UBBRS15 to UBBRS0 bits of this register, set the UBCTL0.UBTXE and UBCTL0.UBRXE bits to 0 or clear the UBCTL0.UBPWR bit to 0.

After reset: FFFFH R/W Address: FFFFFA42H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UBCTL2	UB BRS 15	UB BRS 14	UB BRS 13	UB BRS 12	UB BRS 11	UB BRS 10	UB BRS 9	UB BRS 8	UB BRS 7	UB BRS 6	UB BRS 5	UB BRS 4	UB BRS 3	UB BRS 2	UB BRS 1	UB BRS 0

Remark For the UBBRS15 to UBBRS0 bits, see **Table 15-3 Division Value of 16-bit Counter**.

Table 15-3. Division Value of 16-bit Counter

UB BRS 15	UB BRS 14	UB BRS 13	UB BRS 12	UB BRS 11	UB BRS 10	UB BRS 9	UB BRS 8	UB BRS 7	UB BRS 6	UB BRS 5	UB BRS 4	UB BRS 3	UB BRS 2	UB BRS 1	UB BRS 0	k	Output Clock Selected
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	4	fxx/(2 × k)
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4	fxx/(2 × k)
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5	fxx/(2 × k)
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	6	fxx/(2 × k)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	65532	fxx/(2 × k)
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	65533	fxx/(2 × k)
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	65534	fxx/(2 × k)
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	65535	fxx/(2 × k)

Remarks 1. fxx: Peripheral clock

2. k: Value set by the UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits (k = 4, 5, 6, …, 65535)

3. x: 0 or 1

(4) UARTB transmit data register (UBTX)

The UBTX register is used to set transmit data. It functions as the 8-bit × 1-stage UBTX register, in the single mode (UBFIC0.UBMOD bit = 0), and as the 8-bit × 16-stage transmit FIFO in the FIFO mode (UBFIC0.UBMOD bit = 1).

In the single mode, transmission is started by writing transmit data to the UBTX register when transmission is enabled (UBCTL0.UBTXE bit = 1). When data can be written to the UBTX register (when 1 byte of data is transferred from the UBTX register to the transmit shift register), a transmission enable interrupt request signal (INTUBTIT) is generated.

In the FIFO mode, transmission is started by enabling transmission (UBTXE bit = 1) after writing at least the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less to transmit FIFO. When the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register (transmit data of the number set as the trigger can be written to transmit FIFO), a transmission enable interrupt request signal (INTUBTIT) is generated. In the FIFO mode, a FIFO transmission enable interrupt request signal (INTUBTIF) is generated when there is no more data in transmit FIFO and the transmit shift register (when the FIFO and register become empty).

For the generation timing of the interrupt, see **15.5 Interrupt Request Signals**.

When 7-bit length data is transmitted with the LSB first, bits 6 to 0 of the transmit data register are transmitted as the transmit data from the LSB (bit 0) with the MSB (bit 7) always being 0. When data is transmitted with the MSB first, bits 7 to 1 of the transmit data register are transmitted as the transmit data from the MSB (bit 7) with the LSB (bit 0) always being 0.

This register is write-only in 8-bit units. Data is written to the transmit data register.

Reset sets this register to FFH.

After reset: FFH W Address: FFFFFA48H

	7	6	5	4	3	2	1	0
UBTX	UBTD7	UBTD6	UBTD5	UBTD4	UBTD3	UBTD2	UBTD1	UBTD0

(5) UARTB receive data register AP (UBRXAP), UARTB receive data register (UBRX)

These registers store parallel data converted by the receive shift register. They function as the 8-bit × 1-stage UBRX register, in the single mode (UBFIC0.UBMOD bit = 0), and as the 16-bit × 16-stage receive FIFO (UBRXAP register) in the FIFO mode (UBFIC0.UBMOD bit = 1).

The receive data is stored in the lower 8 bits of the receive FIFO (UBRXAP register) and the error information of the received data is stored in the higher 8 bits (bit 8 and bit 9). If a reception error (such as a parity error or a framing error) occurs in the FIFO mode, the UBRXAP register is read in 16-bit (halfword) units. In this way, the flag of the data stored in receive FIFO can be checked (error information is appended as UBPEF bit = 1 or UBFEF bit = 1), so that the error data can be recognized (when the lower 8 bits of the UBRXAP register are read in 8-bit (byte) units, the higher 8 bits are discarded. Therefore, if no error has occurred, the receive data of the UBRXAP register can be read successively by being read in 8-bit (byte) units in the same way as the UBRX register).

If reception is enabled (UBCTL0.UBRXE bit = 1), the receive data is transferred from the receive shift register to the receive data register, in synchronization with the completion of the shift-in processing of one frame.

By transferring the receive data to the UBRX register in the single mode or by transferring the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to the receive FIFO in the FIFO mode, a reception end interrupt request signal (INTUBTIR) is generated. If data is stored in receive FIFO when the next data does not come (start bit is not detected) even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

For information about the timing for generating these interrupt requests, see **15.5 Interrupt Request Signals**.

If data is received with the LSB first when the data length is specified as 7 bits, the received data is transferred to bits 6 to 0 of the receive data register from the LSB (bit 0), with the MSB (bit 7) always being 0. If data is received with the MSB first, it is transferred to bits 7 to 1 of the receive data register from the MSB (bit 7) with the LSB (bit 0) always being 0. However, if an overrun error occurs, the receive data at that time is not transferred to the receive data register.

The UBRXAP register is read-only in 16-bit units. However, the lower 8 bits of the UBRXAP register are read-only in 8-bit units.

The UBRX register is read-only in 8-bit units.

In addition to reset input, the value of these registers can be set to FFH in the single mode or to 00FFH in the FIFO mode, by clearing the UBCTL0.UBPWR bit to 0.

Cautions 1. The UBPEF and UBFEF bits cannot be read because these registers serve as 8-bit registers in the single mode.

2. When no reception error has occurred in the FIFO mode, the receive data of the UBRXAP register can be read successively by reading the lower 8 bits of the UBRXAP register in 8-bit (byte) units. An 8-bit access to the higher 8 bits is prohibited. If they are accessed, the operation is not guaranteed.

Cautions 3. Do not perform the following operations when debugging a system that uses the single mode.

- **Setting a break for an instruction immediately after the UBRX register is read**
- **Setting a break before DMA transfer with the UBRX register specified as the transfer source is ended**
- **Setting a break before end of reception of the next data after reception of data and reading the UBRX register, and checking the UBRX register in the I/O register window of the debugger**

If any of these operations is performed, an overrun error may occur during the subsequent reception.

After reset: 00FFH R Address: FFFFFA46H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UBRXAP	0	0	0	0	0	0	UBPEF	UBFEF	UBRD7	UBRD6	UBRD5	UBRD4	UBRD3	UBRD2	UBRD1	UBRD0

After reset: FFH R Address: FFFFFA46H

	7	6	5	4	3	2	1	0
UBRX	UBRD7	UBRD6	UBRD5	UBRD4	UBRD3	UBRD2	UBRD1	UBRD0

UBPEF	Parity error flag
0	No parity error
1	Parity error occurs (during reception).
• The UBPEF bit is valid only in the FIFO mode (UBFIC0.UBMOD bit = 1), and is invalid in the single mode (UBFIC0.UBMOD bit = 0). • The operation of the UBPEF bit differs depending on the set values of the UBCTL0.UBPS1 and UBCTL0.UBPS0 bits.	

UBFEF	Framing error flag
0	No framing error
1	Framing error occurs (during reception).
• The UBFEF bit is valid only in the FIFO mode (UBFIC0.UBMOD bit = 1), and is invalid in the single mode (UBFIC0.UBMOD bit = 0). • Only the first bit of the stop bits of the receive data is checked, regardless of the stop bit length.	

UBRD7 to UBRD0	Stores receive data.

(6) UARTB FIFO control register 0 (UBFIC0)

The UBFIC0 register is used to select the operation mode of UARTB and the functions that become valid in the FIFO mode (UBMOD bit = 1). In the FIFO mode, it clears transmit FIFO/receive FIFO and specifies the timing mode in which the transmission enable interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR) is generated.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFFA4AH

	7	6	5	4	3	2	1	0
UBFIC0	UBMOD	0	0	0	UBTFC	UBRFC	UBITM	UBIRM

UBMOD	Specification of UARTB operation mode
0	Single mode
1	FIFO mode

UBTFC	Transmit FIFO clear trigger bit
0	Normal status
1	Clear (This bit automatically returns to 0 after transmit FIFO is cleared.)

- The UBTFC bit is valid only in the FIFO mode (UBMOD bit = 1), and is invalid in the single mode (UBMOD bit = 0).
- When 1 is written to the UBTFC bit, the pointer to transmit FIFO is cleared to 0. In the pending mode (UBITM bit = 0), the interrupt request signal (INTUBTIT) held pending is cleared[Note]. However, bit 7 (UTIF) of the interrupt control register (UTIC) is not cleared to 0. Clear this bit to 0 as necessary.
  When 0 is written to the UBTFC bit, the status is retained. No operation, such as clearing or setting, is executed.
- When writing 1 to the UBTFC bit, be sure to clear the UBCTL0.UBTXE bit to 0 (disabling transmission). If 1 is written to the UBTFC bit when the UBTXE bit is 1 (transmission enabled), the operation is not guaranteed.

Note After transmit FIFO is cleared (UBTFC bit = 1), accessing the registers related to UARTB is prohibited for the duration of four cycles of fxx/2 or until clearing the UBTFC bit (automatic recovery) is confirmed by reading the UBFIC0 register. If these registers are accessed, the operation is not guaranteed.

Remark fxx: Peripheral clock

(2/2)

<table>
<tr><th>UBRFC</th><th>Receive FIFO (UBRXAP) clear trigger bit</th></tr>
<tr><td>0</td><td>Normal status</td></tr>
<tr><td>1</td><td>Clear (This bit automatically returns to 0 after receive FIFO is cleared.)</td></tr>
<tr><td colspan="2">• The UBRFC bit is valid only in the FIFO mode (UBMOD bit = 1), and is invalid in the single mode (UBMOD bit = 0).
• When 1 is written to the UBRFC bit, the pointer to receive FIFO is cleared to 0. In the pending mode (UBIRM bit = 0), the interrupt request signal (INTUBTIR) held pending is cleared^{Note}. However, bit 7 (URIF) of the interrupt control register (URIC) is not cleared to 0. Clear this bit to 0 as necessary.
When 0 is written to the UBRFC bit, the status is retained. No operation, such as clearing or setting, is executed.
• When writing 1 to the UBRFC bit, be sure to clear the UBCTL0.UBRXE bit to 0 (disabling reception). If 1 is written to the UBRFC bit when the UBRXE bit is 1 (reception enabled), the operation is not guaranteed.</td></tr>
</table>

<table>
<tr><th>UBITM</th><th>Specification of INTUBTIT interrupt generation timing in FIFO mode</th></tr>
<tr><td>0</td><td>Pending mode</td></tr>
<tr><td>1</td><td>Pointer mode</td></tr>
<tr><td colspan="2">In the FIFO mode, the INTUBTIT signal is generated as soon as transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register. After the INTUBTIT signal request has been generated, specify the timing of actually generating the INTUBTIT signal as the pending mode or pointer mode. For details, see <b>15.6 (2) Pending mode/pointer mode</b>.</td></tr>
</table>

<table>
<tr><th>UBIRM</th><th>Specification of INTUBTIR interrupt generation timing in FIFO mode</th></tr>
<tr><td>0</td><td>Pending mode</td></tr>
<tr><td>1</td><td>Pointer mode</td></tr>
<tr><td colspan="2">In the FIFO mode, the INTUBTIR signal is generated as soon as receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits have been transferred from the receive shift register to receive FIFO. After the INTUBTIR signal request has been generated, specify the timing of actually generating the INTUBTIR signal as the pending mode or pointer mode. For details, see <b>15.6 (2) Pending mode/pointer mode</b>.</td></tr>
</table>

Note After receive FIFO (UBRXAP) is cleared (UBRFC bit = 1), accessing the registers related to UARTB is prohibited for the duration of four cycles of fxx/2 or until clearing the UBRFC bit (automatic recovery) is confirmed by reading the UBFIC0 register. If these registers are accessed, the operation is not guaranteed.

Remark fxx: Peripheral clock

(7) UARTB FIFO control register 1 (UBFIC1)

The UBFIC1 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It generates a reception timeout interrupt request signal (INTUBTITO) if data is stored in receive FIFO when the next data does not come (start bit is not detected) after the lapse of the time set by the UBTC4 to UBTC0 bits (next data reception wait time), after the stop bit has been received.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFA4BH

	7	6	5	4	3	2	1	0
UBFIC1	UBTCE	0	0	UBTC4	UBTC3	UBTC2	UBTC1	UBTC0

UBTCE	Specification of timeout counter function disable/enable
0	Disable use of timeout counter function.
1	Enable use of timeout counter function.

UBTC4	UBTC3	UBTC2	UBTC1	UBTC0	Next data reception wait time
0	0	0	0	0	32 bytes (32 × 8/baud rate)
0	0	0	0	1	31 bytes (31 × 8/baud rate)
0	0	0	1	0	30 bytes (30 × 8/baud rate)
0	0	0	1	1	29 bytes (29 × 8/baud rate)
• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	0	0	4 bytes (4 × 8/baud rate)
1	1	1	0	1	3 bytes (3 × 8/baud rate)
1	1	1	1	0	2 bytes (2 × 8/baud rate)
1	1	1	1	1	1 byte (1 × 8/baud rate)

When counting up of the reception wait time, set by the UBTC4 to UBTC0 bits, is complete, the count value of the timeout counter is cleared to 0, regardless of the status of the data stored in receive FIFO. When the next start bit is later detected, counting is started again from the stop bit of that data.

(8) UARTB FIFO control register 2 (UBFIC2)

The UBFIC2 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It sets the timing of generating an interrupt, using the number of transmit/receive data as a trigger. When data is transmitted, the number of data transferred from transmit FIFO is specified as the condition of generating the interrupt. When data is received, the number of data stored in receive FIFO is specified as the interrupt generation condition.

This register can be read or written in 16-bit units.

When the higher 8 bits of the UBFIC2 register can be used as the UBFIC2H register and the lower 8 bits, as the UBFIC2L register, these registers can be read or written in 8-bit units.

Reset sets the UBFIC2 register to 0000H and the UBFIC2H and UBFIC2L registers to 00H.

Caution Be sure to set the UBCTL0.UBTXE bit (to disable transmission) and UBCTL0.UBRXE bit (to disable reception) to 0 before writing data to the UBFIC2 register. If data is written to the UBFIC2 register with the UBTXE or UBRXE bit set to 1, the operation is not guaranteed.

(1/2)

After reset: 0000H R/W Address: FFFFFA4CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UBFIC2	0	0	0	0	UBTT3	UBTT2	UBTT1	UBTT0	0	0	0	0	UBRT3	UBRT2	UBRT1	UBRT0

UBTT3	UBTT2	UBTT1	UBTT0	Number of data of transmit FIFO set as trigger	Pointer mode	Pending mode
0	0	0	0	1 byte	Settable	Settable
0	0	0	1	2 bytes	Setting prohibited	
0	0	1	0	3 bytes		
0	0	1	1	4 bytes		
0	1	0	0	5 bytes		
0	1	0	1	6 bytes		
0	1	1	0	7 bytes		
0	1	1	1	8 bytes		
1	0	0	0	9 bytes		
1	0	0	1	10 bytes		
1	0	1	0	11 bytes		
1	0	1	1	12 bytes		
1	1	0	0	13 bytes		
1	1	0	1	14 bytes		
1	1	1	0	15 bytes		
1	1	1	1	16 bytes		

- Set the number of transmit FIFO transmit data to be the trigger.
- Each time data of the specified number has shifted out from transmit FIFO to the transmit shift register, the INTUBTIT signal is generated.
  In the pending mode (UBFIC0.UBITM bit = 0), the INTUBTIT signal is generated under the conditions of the pending mode.
- In the pointer mode (UBFIC0.UBITM bit = 1), the number of transmit data set as the trigger can be only 1 byte (UBTT3 to UBTT0 bits = 0000), and other settings are prohibited. If a setting of other than 1 byte is made, the operation is not guaranteed.

(2/2)

UBRT3	UBRT2	UBRT1	UBRT0	Number of data of transmit FIFO set as trigger	Pointer mode	Pending mode
0	0	0	0	1 byte	Settable	Settable
0	0	0	1	2 bytes	Setting prohibited	
0	0	1	0	3 bytes		
0	0	1	1	4 bytes		
0	1	0	0	5 bytes		
0	1	0	1	6 bytes		
0	1	1	0	7 bytes		
0	1	1	1	8 bytes		
1	0	0	0	9 bytes		
1	0	0	1	10 bytes		
1	0	1	0	11 bytes		
1	0	1	1	12 bytes		
1	1	0	0	13 bytes		
1	1	0	1	14 bytes		
1	1	1	0	15 bytes		
1	1	1	1	16 bytes		

- Set the number of receive FIFO receive data to be the trigger.
- Each time data of the specified number has been stored from the receive shift register to receive FIFO, the INTUBTIR interrupt is generated.
  In the pending mode (UBFIC0.UBIRM bit = 0), the INTUBTIR signal is generated under the conditions of the pending mode.
- In the pointer mode (UBFIC0.UBIRM bit = 1), the number of receive data set as the trigger can be only 1 byte (UBRT3 to UBRT0 bits = 0000), and other settings are prohibited. If a setting of other than 1 byte is made, the operation is not guaranteed.

(9) UARTB FIFO status register 0 (UBFIS0)

The UBFIS0 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It is used to read the number of bytes of the data stored in receive FIFO.

This register is read-only in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R Address: FFFFFA4EH

	7	6	5	4	3	2	1	0
UBFIS0	0	0	0	UBRB4	UBRB3	UBRB2	UBRB1	UBRB0

UBRB4	UBRB3	UBRB2	UBRB1	UBRB0	Receive FIFO pointer
0	0	0	0	0	0 bytes
0	0	0	0	1	1 byte
0	0	0	1	0	2 bytes
0	0	0	1	1	3 bytes
0	0	1	0	0	4 bytes
0	0	1	0	1	5 bytes
0	0	1	1	0	6 bytes
0	0	1	1	1	7 bytes
0	1	0	0	0	8 bytes
0	1	0	0	1	9 bytes
0	1	0	1	0	10 bytes
0	1	0	1	1	11 bytes
0	1	1	0	0	12 bytes
0	1	1	0	1	13 bytes
0	1	1	1	0	14 bytes
0	1	1	1	1	15 bytes
1	0	0	0	0	16 bytes
Other than above					Invalid
Indicates the number of bytes (readable bytes) of the data stored in receive FIFO as a receive FIFO pointer.					

(10) UARTB FIFO status register 1 (UBFIS1)

The UBFIS1 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). This register can be used to read the number of empty bytes of transmit FIFO.

This register is read-only in 8-bit units.

Reset sets this register to 10H.

Caution The values of the UBTB4 to UBTB0 bits are reflected after transmit data has been written to the UBTX register and then time of two cycles of the fxx/2 has passed. Therefore, care must be exercised when referencing the UBFIS1 register after transmit data has been written to the UBTX register.

After reset: 10H R Address: FFFFFA4FH

	7	6	5	4	3	2	1	0
UBFIS1	0	0	0	UBTB4	UBTB3	UBTB2	UBTB1	UBTB0

UBTB4	UBTB3	UBTB2	UBTB1	UBTB0	Transmit FIFO pointer
0	0	0	0	0	0 bytes
0	0	0	0	1	1 byte
0	0	0	1	0	2 bytes
0	0	0	1	1	3 bytes
0	0	1	0	0	4 bytes
0	0	1	0	1	5 bytes
0	0	1	1	0	6 bytes
0	0	1	1	1	7 bytes
0	1	0	0	0	8 bytes
0	1	0	0	1	9 bytes
0	1	0	1	0	10 bytes
0	1	0	1	1	11 bytes
0	1	1	0	0	12 bytes
0	1	1	0	1	13 bytes
0	1	1	1	0	14 bytes
0	1	1	1	1	15 bytes
1	0	0	0	0	16 bytes
Setting prohibited					Invalid
Indicates the number of empty bytes of transmit FIFO (bytes that can be written) as a transmit FIFO pointer.					

15.5 Interrupt Request Signals

The following five types of interrupt requests are generated from UARTB.

- Reception error interrupt request signal (INTUBTIRE)
- Reception end interrupt request signal (INTUBTIR)
- Transmission enable interrupt request signal (INTUBTIT)
- FIFO transmission end interrupt request signal (INTUBTIF)
- Reception timeout interrupt request signal (INTUBTITO)

The default priorities among these five types of interrupt requests is, from high to low, reception error interrupt request signal, reception end interrupt request signal, transmission enable interrupt request signal, FIFO transmission end interrupt request signal, and reception timeout interrupt request signal.

Table 15-4. Generated Interrupts and Default Priorities

Interrupt	Priority
Reception error	1
Reception end	2
Transmission enable	3
FIFO transmission end	4
Reception timeout	5

(1) Reception error interrupt request signal (INTUBTIRE)

(a) Single mode

When reception is enabled, a reception error interrupt request signal is generated according to the logical OR of the three types of reception errors (parity error, framing error, overrun error) explained for the UBSTR register.

When reception is disabled, no reception error interrupt request signal is generated.

(b) FIFO mode

When reception is enabled, a reception error interrupt request signal is generated according to the logical OR of the three types of reception errors (parity error, framing error, overflow error) explained for the UBSTR register.

When reception is disabled, no reception error interrupt request signal is generated.

(2) Reception end interrupt request signal (INTUBTIR)

(a) Single mode

When reception is enabled, a reception end interrupt request signal is generated if data is shifted into the receive shift register and stored in the UBRX register (if the receive data can be read).

When reception is disabled, no reception end interrupt request signal is generated.

(b) FIFO mode

When reception is enabled, a reception end interrupt request signal is generated if data is shifted into the receive shift register and receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is transferred to receive FIFO (if receive data of the specified number can be read).

When reception is disabled, no reception end interrupt request signal is generated.

(3) Transmission enable interrupt request signal (INTUBTIT)

(a) Single mode

The transmission enable interrupt request signal is generated if transmit data of one frame, including 7 or 8 bits of characters, is shifted out from the transmit shift register and the UBTX register becomes empty (if transmit data can be written).

(b) FIFO mode

The transmission enable interrupt request signal is generated if transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred to the transmit shift register from transmit FIFO (if transmit data of the specified number can be written).

(4) FIFO transmission end interrupt request signal (INTUBTIF)

(a) Single mode

Cannot be used.

(b) FIFO mode

The FIFO transmission end interrupt request signal is generated when no more data is in transmit FIFO and the transmit shift register (when the FIFO and register become empty). After the FIFO transmission end interrupt request signal has occurred, clear the interrupt request signal (INTUBTIT) held pending in the pending mode (UBFIC0.UBITM bit = 0) by clearing the FIFO (UBFIC0.UBTFC bit = 1).

Caution **If the FIFO transmission end interrupt request signal is generated (all transmit data are not transmitted) because writing the next transmit data to transmit FIFO is delayed, do not clear the FIFO.**

(5) Reception timeout interrupt request signal (INTUBTITO)

(a) Single mode

Cannot be used.

(b) FIFO mode

The reception timeout interrupt request signal is generated if data is stored in receive FIFO when the next data does not come (start bit is not detected) even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed, when the timeout counter function is used (UBFIC1.UBTCE bit = 1).

The reception timeout interrupt request signal is not generated while reception is disabled.

If receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is not received, the timing of reading the number of receive data less than the specified number can be set by the reception timeout interrupt request signal.

Since the timeout counter starts counting at start bit detection, a receive timeout interrupt request signal does not occur if data of 1 character has not been received.

15.6 Control Modes

(1) Single mode/FIFO mode

The single mode or FIFO mode can be selected by using the UBFIC0.UBMOD bit.

(a) Single mode

- Each of the UBRX and UBTX registers consists of 8 bits × 1 stage.
- When 1 byte of data is received, the INTUBTIR signal is generated.
- If the next reception operation of UARTB is ended before the receive data of the UBRX register is read after the INTUBTIR signal has been generated, the INTUBTIRE signal is generated and an overrun error occurs.

(b) FIFO mode

- Receive FIFO (UBRXAP register) consists of 16 bits × 16 stages and transmit FIFO consists of 8 bits × 16 stages.
- Receive FIFO can recognize error data by reading the 16-bit UBRXAP register only when a reception error (parity error or framing error) occurs.
- Transmission is started when transmission is enabled (UBCTL0.UBTXE bit = 1) after transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less are written to transmit FIFO.
- The pending mode or pointer mode can be selected for the generation timing of the INTUBTIT and INTUBTIR signals.

(2) Pending mode/pointer mode

The pending mode or pointer mode can be selected by using the UBFIC0.UBITM and UBFIC0.UBIRM bits in the FIFO mode (UBFIC0.UBMOD bit = 1).

If transmission is started by writing data of more than double the amount set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to transmit FIFO, the transmission enable interrupt request signal (INTUBTIT) may occur more than once. The reception end interrupt request signal (INTUBTIR) may also occur more than once if the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is 8 bytes or less in receive FIFO. In the pending or pointer mode, it can be specified how an interrupt is handled after it has been held pending.

(a) Pending mode

(i) During transmission (writing to transmit FIFO)

- If the data of the first transmission enable interrupt request signal (INTUBTIT) is not written to transmit FIFO after the interrupt has occurred, the second INTUBTIT signal does not occur (is held pending) even if the generation condition of the second INTUBTIT signal is satisfied (when transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred from transmit FIFO to the transmit shift register).

  When data for the first INTUBTIT signal is later written to transmit FIFO, the pending INTUBTIT signal is generated[Note].

  Note The number of pending interrupts is as follows.

  When trigger is set to 1 byte (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0000): 15 times max.
  When trigger is set to 2 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0001): 7 times max.
  :
  When trigger is set to 6 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0101): 1 time max.
  When trigger is set to 7 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0110): 1 time max.
  When trigger is set to 8 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0111): 1 time max.

- In the pending mode, transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is always written to transmit FIFO when the transmission enable interrupt request signal (INTUBTIT) occurs. Writing data to transmit FIFO is prohibited if the data is more or less than the specified number. If data more or less than the specified number is written, the operation is not guaranteed.
- Fix the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 0000 (set number of transmit data: 1 byte) to write transmit data to transmit FIFO by DMA. If any other setting is made, the operation is not guaranteed.

(ii) During reception (reading from receive FIFO)

- If data for the first reception end interrupt request signal (INTUBTIR) is not read from receive FIFO, the second INTUBTIR signal does not occur (is held pending) even if the generation condition of the second INTUBTIR is satisfied (if receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits can be read from receive FIFO). When data for the first INTUBTIR signal is later read from the receive FIFO, the pending INTUBTIR signal is generated[Note].

Note The number of pending interrupts is as follows.

When trigger is set to 1 byte (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0000): 15 times max.
When trigger is set to 2 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0001): 7 times max.
:
When trigger is set to 6 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0101): 1 time max.
When trigger is set to 7 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0110): 1 time max.
When trigger is set to 8 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0111): 1 time max.

- In the pending mode, receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is always read from receive FIFO when the reception end interrupt request signal (INTUBTIR) occurs. Reading data from receive FIFO is prohibited if the data is more or less than the specified number. If data more or less than the specified number is read, the operation is not guaranteed.
- Fix the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to 0000 (set number of receive data: 1 byte) to read receive data from receive FIFO by DMA. If any other setting is made, the operation is not guaranteed.

(b) Pointer mode

(i) During transmission (writing to transmit FIFO)

- Each time the data of 1 byte is transferred to the transmit shift register from transmit FIFO, a transmission enable interrupt request signal (INTUBTIT) occurs.
- In the pointer mode, be sure to fix the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 0000 (set number of transmit data: 1 byte) as the number of transmit data set as the trigger for transmit FIFO when the transmission enable interrupt request signal (INTUBTIT) occurs. If any other setting is made, the operation is not guaranteed.
- Writing transmit data to transmit FIFO by DMA is prohibited. The operation is not guaranteed if DMA control is used.
- After the transmission enable interrupt request signal (INTUBTIT) has been acknowledged, data of the number of empty bytes of transmit FIFO can be written to transmit FIFO by referencing the UBFIS1 register.

(ii) During reception (reading from receive FIFO)

- Each time the data of 1 byte is transferred to receive FIFO from the receive shift register, a reception end interrupt request signal (INTUBTIR) occurs.
- In the pointer mode, be sure to fix the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to 0000 (set number of receive data: 1 byte) as the number of receive data set as the trigger for receive FIFO when the reception end interrupt request signal (INTUBTIR) occurs. If any other setting is made, the operation is not guaranteed.
- Reading receive data from receive FIFO by DMA is prohibited. The operation is not guaranteed if DMA control is used.
- After the reception end interrupt request signal (INTUBTIR) has been acknowledged, data of the number of bytes stored in receive FIFO can be read from receive FIFO by referencing the UBFIS0 register. In some cases, however, data is not stored in receive FIFO even though the INTUBTIR signal is generated (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000). In these cases, do not read data from receive FIFO. Always read data from receive FIFO when the number of bytes stored in receive FIFO is 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).

15.7 Operation

15.7.1 Data format

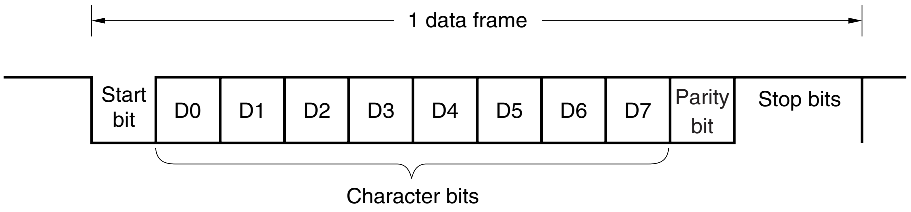
Full-duplex serial data transmission and reception can be performed.

The transmit/receive data format consists of one data frame containing a start bit, character bits, a parity bit, and stop bits as shown in Figure 15-3.

The character bit length within one data frame, the type of parity, and the stop bit length are specified by UARTB control register 0 (UBCTL0).

Also, data is transferred with LSB first/MSB first.

Figure 15-3. Asynchronous Serial Interface Transmit/Receive Data Format (LSB-First Transfer)



- Start bit ··· 1 bit
- Character bits ··· 7 bits or 8 bits
- Parity bit ··· Even parity, odd parity, 0 parity, or no parity
- Stop bits ··· 1 bit or 2 bits

15.7.2 Transmit operation

In the single mode (UBFIC0.UBMOD bit = 0), transmission is enabled when the UBCTL0.UBTXE bit is set to 1, and transmission is started when transmit data is written to the UBTX register.

In the FIFO mode (UBFIC0.UBMOD bit = 1), transmission is started when transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO and then the UBTXE bit is set to 1.

Caution Setting the UBCTL0.UBTXE bit to 1 before writing transmit data to transmit FIFO in the FIFO mode is prohibited. The operation is not guaranteed if this setting is made.

(1) Transmission enabled state

This state is set by the UBCTL0.UBTXE bit.

- UBTXE = 1: Transmission enabled state
- UBTXE = 0: Transmission disabled state

However, because this bit is also used by CSIF2, enable transmission after setting the CF2CTL0.CF2PWR bit to 0.

Since UARTB does not have a CTS (transmission enabled signal) input pin, a port should be used to confirm whether the destination is in the reception enabled state.

(2) Starting a transmit operation

- **In single mode (UBFIC0.UBMOD bit = 0)**
  In the single mode, transmission is started when transmit data is written to the UBTX register while transmission is enabled.
- **In FIFO mode (UBFIC0.UBMOD bit = 1)**
  In the FIFO mode, transmission is started when transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO and then transmission is enabled (UBTXE bit = 1).

Data in the transmit data register (UBTX register in single mode or transmit FIFO in the FIFO mode) is transferred to the transmit shift register when transmission is started. Then, the transmit shift register outputs data to the TXDB pin sequentially beginning with the LSB (the transmit data is transferred sequentially starting with the start bit). The start bit, parity bit, and stop bits are added automatically.

(3) Transmission interrupt request signal

(a) Transmission enable interrupt request signal (INTUBTIT)

- **In single mode (UBFIC0.UBMOD bit = 0)**
  In the single mode, the transmission enable interrupt request signal (INTUBTIT) occurs when transmit data can be written to the UBTX register (when 1 byte of data is transferred from the UBTX register to the transmit shift register).

- **In FIFO mode (UBFIC0.UBMOD bit = 1)**
  In the FIFO mode, the INTUBTIT signal occurs when transmit data of the number set as the trigger specified by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred from transmit FIFO to the transmit shift register (if transmit data of the number set as the trigger can be written).

- **If pending mode is specified (UBFIC0.UBITM bit = 0) in FIFO mode**
  If the pending mode is specified in the FIFO mode, the second INTUBTIT signal is held pending after the first INTUBTIT signal has occurred, until as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits are written to transmit FIFO, even if the generation condition of the second INTUBTIT signal is satisfied. When as many transmit data as the number set as the trigger are written to transmit FIFO in response to the first INTUBTIT signal, the second pending INTUBTIT signal is generated.

- **If pointer mode is specified (UBFIC0.UBITM bit = 1) in FIFO mode**
  If the pointer mode is specified in the FIFO mode, the second INTUBTIT signal occurs when the generation condition of the second INTUBTIT signal is satisfied even if as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits are not written to transmit FIFO when the first INTUBTIT signal occurs.

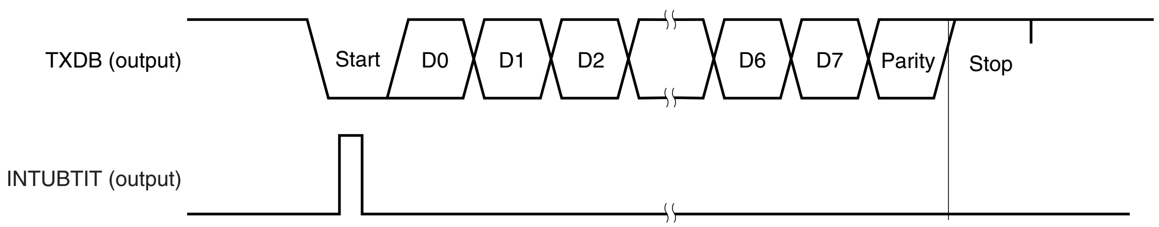
(b) FIFO transmission end interrupt request signal (INTUBTIF)

The FIFO transmission end interrupt request signal (INTUBTIF) occurs when no more data is in transmit FIFO and the transmit shift register in the FIFO mode (UBFIC0.UBMOD bit = 1). After the INTUBTIF signal has occurred, clear the pending INTUBTIT signal in the pending mode (UBFIC0.UBITM bit = 0) by clearing the FIFO (UBFIC0.UBTFC bit = 1). If the INTUBTIF signal occurs because writing the next transmit data to transmit FIFO is delayed (if all transmit data have not been transmitted), do not clear the FIFO.

If the data to be transmitted next has not been written to the transmit data register, the transmit operation is suspended.

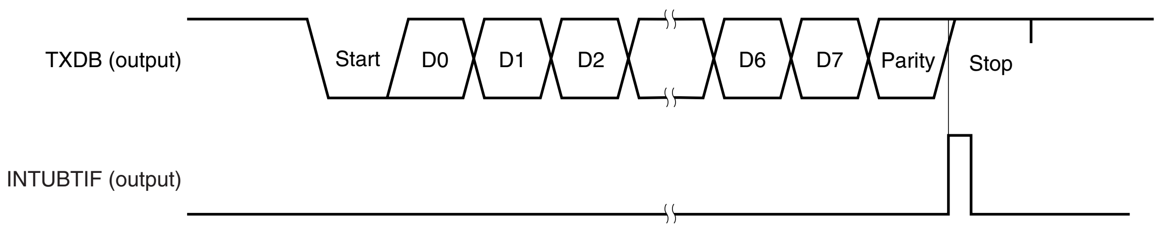
Caution In the single mode, the transmission enable interrupt request signal (INTUBTIT) occurs when the UBTX register becomes empty (when 1 byte of data is transferred from the UBTX register to the transmit shift register). In the FIFO mode, the FIFO transmission end interrupt request signal (INTUBTIF) occurs when data is no longer in transmit FIFO and the transmit shift register (when the FIFO and register are empty). However, the INTUBTIT signal or INTUBTIF signal is not generated if the transmit data register becomes empty due to $\overline{\text{RESET}}$ input.

Figure 15-4. Timing of Asynchronous Serial Interface Transmission Enable Interrupt Request Signal (INTUBTIT)



Remark In the FIFO mode, the INTUBTIT signal occurs at the above timing when as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits are serially transferred.

Figure 15-5. Timing of Asynchronous Serial Interface FIFO Transmission End Interrupt Request Signal (INTUBTIF)



Remark The INTUBTIF signal occurs at the above timing when data is no longer in transmit FIFO and the transmit shift register (when the FIFO and register are empty).

15.7.3 Continuous transmission operation

- **In single mode (UBFIC0.UBMOD bit = 0)**

  In the single mode, the next data can be written to the UBTX register as soon as the transmit shift register has started a shift operation. The timing of transfer can be identified by the transmission enable interrupt request signal (INTUBTIT). By writing the next transmit data to the UBTX register via the INTUBTIT signal within one data frame transmission period, data can be transmitted without an interval and an efficient communication rate can be realized.

  Caution Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing. If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

- **If pending mode is specified (UBFIC0.UBITM bit = 0) in FIFO mode**

  If transmit data of at least the number set as the transmit trigger by UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO, transmission starts.

  If the pending mode is specified in the FIFO mode, as many of the next transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits can be written to transmit FIFO as soon as the transmit shift register has started shifting the last data of the specified number of data. The timing of transfer can be identified by the INTUBTIT signal. By writing as many of the next transmit data as the number set as the trigger to transmit FIFO or writing the data to the FIFO within the transmission period of the data in transmit FIFO via the INTUBTIT signal, data can be transmitted without an interval and an efficient communication rate can be realized.

  Caution Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed. To write transmit data to transmit FIFO by DMA, set the number of transmit data specified as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 1 byte; otherwise the operation will not be guaranteed.

- **If pointer mode is specified (UBFIC0.UBITM bit = 1) in FIFO mode**

  If the pointer mode is specified in the FIFO mode, a INTUBTIT signal occurs and the next data can be written to transmit FIFO as soon as the transmit shift register has started shifting the number of transmit data set as the trigger. At this time, as many data as the number of empty bytes of transmit FIFO can be written by referencing the UBFIS1 register. The timing of transfer can be identified by the INTUBTIT signal. By writing as many of the next transmit data as the number specified as the trigger to transmit FIFO or writing the data to the FIFO within the transmission period of the data in transmit FIFO via the INTUBTIT signal, data can be transmitted without an interval and an efficient communication rate can be realized.

  Caution Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

15.7.4 Receive operation

The awaiting reception state is set by setting the UBCTL0.UBPWR bit to 1 and then setting the UBCTL0.UBRXE bit to 1. RXDB pin sampling begins and a start bit is detected. When the start bit is detected, the receive operation begins, and data is stored sequentially in the receive shift register according to the baud rate that was set.

In the single mode (UBFIC0.UBMOD bit = 0), a reception end interrupt request signal (INTUBTIR) is generated each time the reception of one frame of data is completed. Normally, the receive data is transferred from the UBRX register to memory by this interrupt servicing.

In the FIFO mode (UBFIC0.UBMOD bit = 1), the INTUBTIR signal occurs when as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are transferred to receive FIFO.

If the pending mode is specified (UBFIC0.UBIRM bit = 0) in the FIFO mode, as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits can be read from receive FIFO.

If the pointer mode is specified (UBFIC0.UBIRM bit = 1) in the FIFO mode, as many data as the number of bytes stored in receive FIFO (0 bytes or more) can be read from receive FIFO by referencing the number of receive data specified as the trigger by the UBRT3 to UBRT0 bits (1 byte) or the UBFIS0 register.

Caution If the pointer mode is specified in the FIFO mode and if as many data as the number of bytes stored in receive FIFO are read by referencing the UBFIS0 register, no data may be stored in receive FIFO (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000) even though the reception end interrupt request signal (INTUBTIR) has occurred. In this case, do not read data from receive FIFO. Be sure to read data from receive FIFO after confirming that the number of bytes stored in receive FIFO = 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).

(1) Reception enabled state

This state is set by the UBCTL0.UBRXE bit.

- UBRXE = 1: Reception enabled state
- UBRXE = 0: Reception disabled state

However, because this bit is also used by CSIF2, enable reception after setting the CF2CTL0.CF2PWR bit to 0 and disabling the CSIF2 operation.

In the reception disabled state, the reception hardware stands by in the initial state. At this time, the reception end interrupt request signal or reception error interrupt request signal does not occur, and the contents of the receive data register (UBRX register in the single mode or receive FIFO in the FIFO mode (UBRXAP register)) are retained.

(2) Starting a receive operation

A receive operation is started by the detection of a start bit.

The RXDB pin is sampled using the serial clock from UARTB control register 2 (UBCTL2).

(3) Reception interrupt request signal

(a) Reception end interrupt request signal (INTUBTIR)

- **In single mode (UBFIC0.UBMOD bit = 0)**

  When UBCTL0.UBRXE bit = 1 and the reception of one frame of data is ended (the stop bit is detected) in the single mode, a reception end interrupt request signal (INTUBTIR) is generated and the receive data in the receive shift register is transferred to the UBRX register at the same time.

  Also, if an overrun error occurs, the receive data at that time is not transferred to the UBRX register, and a reception error interrupt request signal (INTUBTIRE) is generated.

  If a parity error or framing error occurs during the reception operation, the reception operation continues up to the position at which the stop bit is received. After completion of reception, an INTUBTIRE signal occurs (the receive data in the receive shift register is transferred to the UBRX register).

  If the UBRXE bit is reset (0) during a receive operation, the receive operation is immediately stopped. At this time, the contents of the UBRX register remain unchanged, the contents of the UARTB status register (UBSTR) are cleared, and the INTUBTIR and INTUBTIRE signals do not occur.

  No INTUBTIR signal is generated when the UBRXE bit = 0 (reception is disabled).

- **In FIFO mode (UBFIC0.UBMOD bit = 1)**

  In the FIFO mode, the reception end interrupt request signal (INTUBTIR) occurs when data of one frame has been received (stop bit is detected) and when as many receive data as the number specified as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are transferred from the receive shift register to receive FIFO. If an overflow error occurs, the receive data is not transferred to receive FIFO and the reception error interrupt request signal (INTUBTIRE) occurs.

  If a parity error or framing error occurs during reception, reception continues up to the reception position of the stop bit. After reception has been completed, the INTUBTIRE signal occurs and the receive data in the receive shift register is transferred to receive FIFO. At this time, error information is appended as the UBRXAP.UBPEF or UBRXAP.UBFEF bit = 1. If the INTUBTIRE signal occurs, the error data can be recognized by reading receive FIFO as a 16-bit register, UBRXAP.

(b) Reception timeout interrupt request signal (INTUBTITO) (only in FIFO mode)

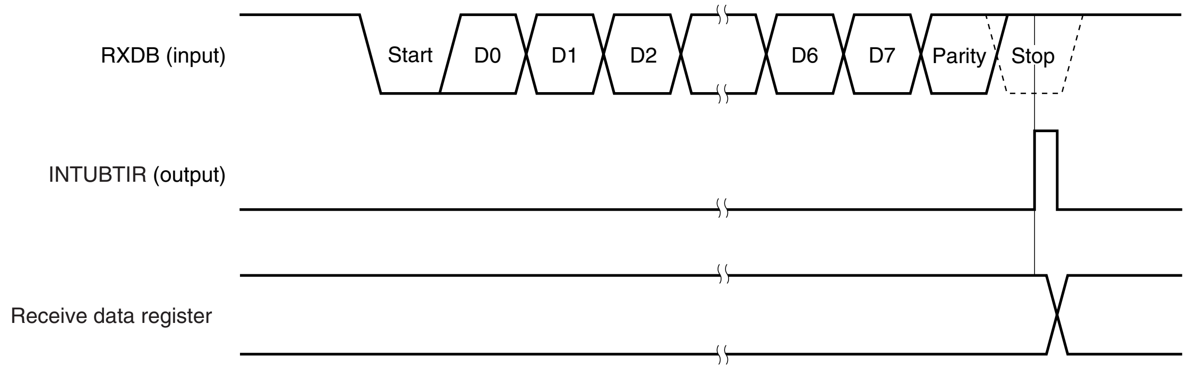
When the timeout counter function (UBFIC1.UBTCE bit = 1) is used in the FIFO mode, the reception timeout interrupt request signal (INTUBTITO) occurs if the next data does not come even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed and if data is stored in receive FIFO.

The INTUBTITO signal does not occur while reception is disabled.

If as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are not received, the timing of reading less receive data than the specified number can be set by the INTUBTITO signal.

Since the timeout counter starts counting at start bit detection, a receive timeout interrupt request signal does not occur if data of 1 character has not been received.

Figure 15-6. Timing of Asynchronous Serial Interface Reception End Interrupt Request Signal (INTUBTIR)



Cautions 1. Be sure to read all the data (the number of data indicated by the UBFIS0.UBRB4 to UBFIS0.UBRB0 bits) stored in the receive data register (UBRX register in the single mode or receive FIFO in the FIFO mode (UBRXAP register)) even when a reception error occurs.

Unless the receive data register is read, an overrun error occurs when the next data is received, causing the reception error status to persist.

If the pending mode is specified in the FIFO mode, however, be sure to clear the FIFO (UBFIC0.UBRFC bit = 1) after reading the data stored in receive FIFO.

In the FIFO mode, the FIFO can be cleared even without reading the data stored in receive FIFO.

If a parity error or framing error occurs in the FIFO mode, the UBRXAP register can be read in 16-bit (halfword) units.

2. Data is always received with one stop bit (1).

A second stop bit is ignored.

15.7.5 Reception error

In the single mode (UBFIC0.UBMOD bit = 0), the three types of errors that can occur during a receive operation are a parity error, framing error, and overrun error. In the FIFO mode (UBFIC0.UBMOD bit = 1), the three types of errors that can occur during a receive operation are a parity error, framing error, and overflow error.

As a result of data reception, the UBSTR.UBPE, UBSTR.UBFE, or UBSTR.UBOVE bit is set to 1 if a parity error, framing error, or overrun error occurs in the single mode. The UBSTR.UBOVF bit is set to 1 if an overflow error occurs in the FIFO mode. The UBRXAP.UBPEF or UBRXAP.UBFEF bit is set to 1 if a parity error or framing error occurs in the FIFO mode. At the same time, a reception error interrupt request signal (INTUBTIRE) occurs. The contents of the error can be detected by reading the contents of the UBSTR or UBRXAP register.

The contents of the UBSTR register are reset when 0 is written to the UBOVF, UBPE, UBFE, or UBOVE bit, or the UBCTL0.UBPWR or UBCTL0.UBRXE bit. The contents of the UBRXAP register are reset when 0 is written to the UBCTL0.UBPWR bit.

Table 15-5. Reception Error Causes

Error Flag	Valid Operation Mode	Error Flag	Reception Error	Cause
UBPE	Single mode	UBPE	Parity error	The parity specification during transmission does not match the parity of the receive data
UBFE		UBFE	Framing error	No stop bit detected
UBOVE		UBOVE	Overrun error	The reception of the next data is ended before data is read from the UBRX register
UBOVF	FIFO mode	UBOVF	Overflow error	The reception of the next data is ended while receive FIFO is full and before data is read.
UBPEF		UBPEF	Parity error	The parity specification during transmission does not match the parity of the data to be received.
UBFEF		UBFEF	Framing error	The stop bit is not detected when the target data is loaded.

15.7.6 Parity types and corresponding operation

A parity bit is used to detect a bit error in communication data. Normally, the same type of parity bit is used at the transmission and reception sides.

(1) Even parity

(a) During transmission

The parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is even. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 1
- If the number of bits with the value “1” within the transmit data is even: 0

(b) During reception

The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is odd.

(2) Odd parity

(a) During transmission

In contrast to even parity, the parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is odd. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 0
- If the number of bits with the value “1” within the transmit data is even: 1

(b) During reception

The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is even.

(3) 0 parity

During transmission the parity bit is set to “0” regardless of the transmit data.

During reception, no parity bit check is performed. Therefore, no parity error is generated regardless of whether the parity bit is “0” or “1”.

(4) No parity

No parity bit is added to the transmit data.

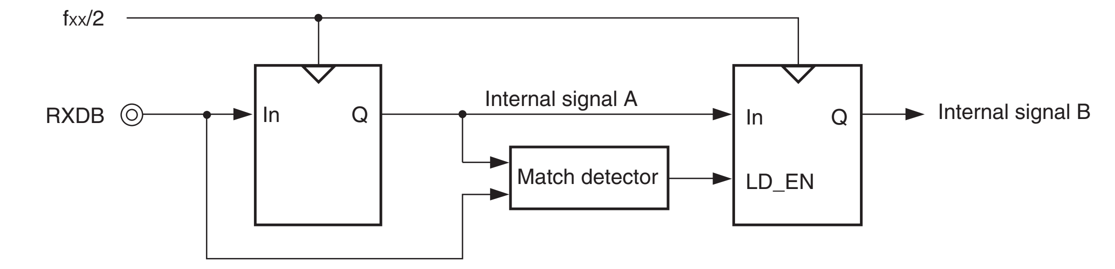
During reception, the receive operation is performed as if there were no parity bit. Since there is no parity bit, no parity error is generated.

15.7.7 Receive data noise filter

The RXDB signal is sampled at the rising edge of input clock fxx/2. If the same sampling value is obtained twice, the match detector output changes, and this output is sampled as input data. Therefore, data not exceeding one clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 15-8**).

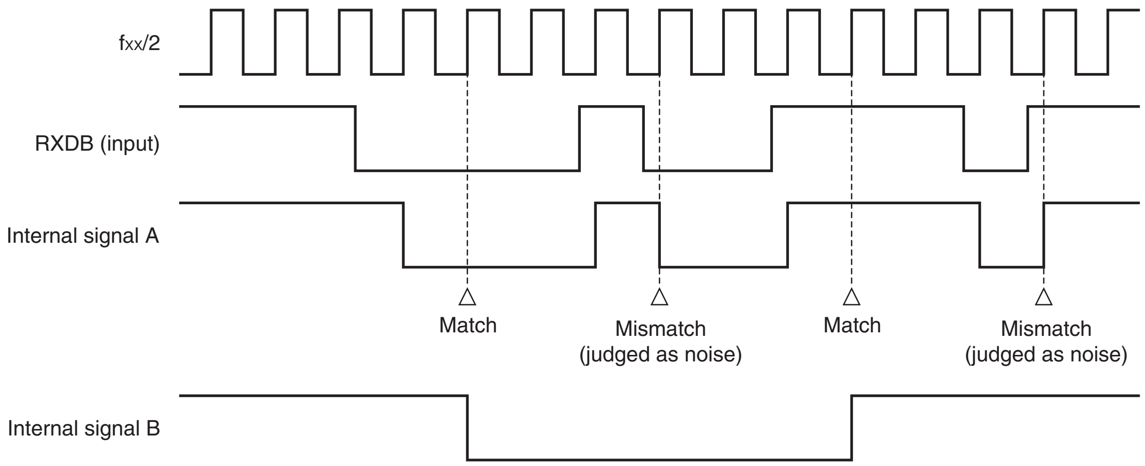
Also, since the circuit is configured as shown in Figure 15-7, internal processing during a receive operation is delayed by up to 2 clocks according to the external signal status.

Figure 15-7. Noise Filter Circuit



Remark fxx: Peripheral clock

Figure 15-8. Timing of RXDB Signal Judged as Noise



Remark fxx: Peripheral clock

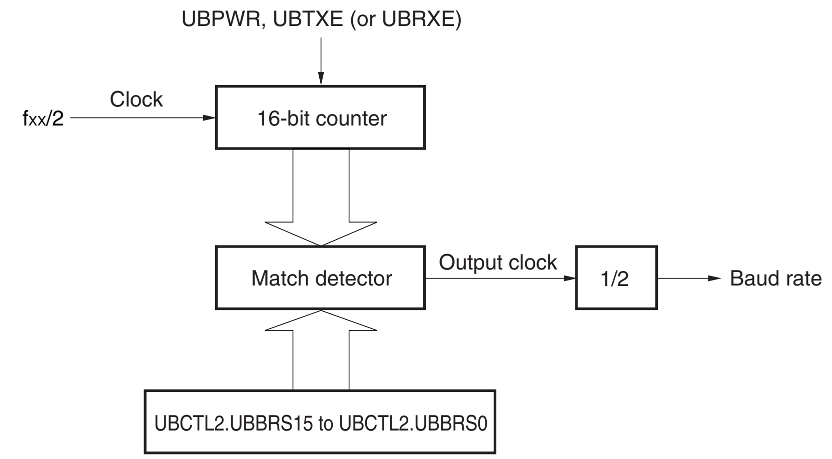
15.8 Dedicated Baud Rate Generator (BRG)

A dedicated baud rate generator, which consists of a 16-bit programmable counter, generates serial clocks during transmission/reception in UARTB. The dedicated baud rate generator output can be selected as the serial clock for each channel.

Separate 16-bit counters exist for transmission and for reception. The baud rate for transmission/reception is the same at the same channel.

(1) Baud rate generator configuration

Figure 15-9. Baud Rate Generator Configuration



Remark fxx: Peripheral clock

(a) Base clock (Clock)

When UBCTL0.UBPWR bit = 1, input clock (fxx/2) is supplied to the transmission/reception unit. This clock is called the base clock. When the UBPWR bit = 0, the clock signal is fixed at low level.

(2) Serial clock generation

A serial clock can be generated according to the settings of the UBCTL2 register.

The 16-bit counter divisor value can be selected according to the UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits.

(a) Baud rate

The baud rate is the value obtained according to the following formula.

$$\text{Baud rate} = \frac{\text{Base clock frequency}}{2 \times k} \text{ [bps]}$$

Base clock frequency = fxx/2 (fxx: peripheral clock)

k = Value set according to UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits (k = 4, 5, 6, ..., 65535)

(b) Baud rate error

The baud rate error is obtained according to the following formula.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. Make sure that the baud rate error during transmission does not exceed the allowable error of the reception destination.

2. Make sure that the baud rate error during reception is within the allowable baud rate range during reception, which is described in paragraph (4).

Example: Base clock (fxx) = 100 MHz = 100,000,000 Hz

Settings of UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits = 0000001010001011B
(k = 651)

Target baud rate = 38,400 bps

Baud rate = 100 M/2/(2 × 65)
= 100,000,000/2/ (2 × 65) = 38,402.45 [bps]

Error = (38,402.45/312,500 − 1) × 100
= 0.0064 [%]

When base clock (fxx) = 100 MHz and k = 80, the error is 0%.

(3) Baud rate setting example

Table 15-6. Baud Rate Generator Setting Data

Baud Rate (bps)	fxx = 100 MHz			fxx = 96 MHz			fxx = 80 MHz		
	k (Decimal)	k (Hexadecimal)	ERR	k (Decimal)	k (Hexadecimal)	ERR	k (Decimal)	k (Hexadecimal)	ERR
300	–	–	–	–	–	–	–	–	–
600	41,667	A2C3	–0.0008	40,000	9C40	0.000	33,333	8235	0.0010
1,200	20,833	5161	0.0016	20,000	4E20	0.000	16,667	411B	–0.0020
2,400	10,417	28B1	–0.0032	10,000	2710	0.000	8,333	208D	0.0040
4,800	5,208	1458	0.0064	5,000	1388	0.000	4,166	1046	0.0160
9,600	2,604	0A2C	0.0064	2,500	09C4	0.000	2,083	0823	0.0160
19,200	1,302	0516	0.0064	1,250	04E2	0.000	1,042	0412	–0.0320
31,250	800	0320	0.0000	768	0300	0.000	640	0280	0.0000
38,400	651	028B	0.0064	625	0271	0.000	521	0209	–0.0320
76,800	326	0146	–0.147	313	0139	–0.1597	260	0104	0.1603
153,600	163	00A3	–0.147	156	009C	0.1603	130	0082	0.1603
312,500	80	0050	0.0000	77	004D	–0.2597	64	0040	0.0000
500,000	50	0032	0.0000	48	0030	0.000	40	0028	0.0000
1,000,000	25	0019	0.0000	24	0018	0.000	20	0014	0.0000
2,000,000	13	000D	–3.8462	12	000C	0.000	10	000A	0.0000
3,000,000	8	0008	4.1667	8	0008	0.000	7	0007	–4.7619
4,000,000	6	0006	4.1667	6	0006	0.000	5	0005	0.0000
5,000,000	5	0005	0.0000	5	0005	–4.0000	4	0004	0.0000

Caution **The maximum allowable frequency of the peripheral clock (fxx) is 100 MHz.**
The maximum transfer speed of the baud rate is 5 Mbps.

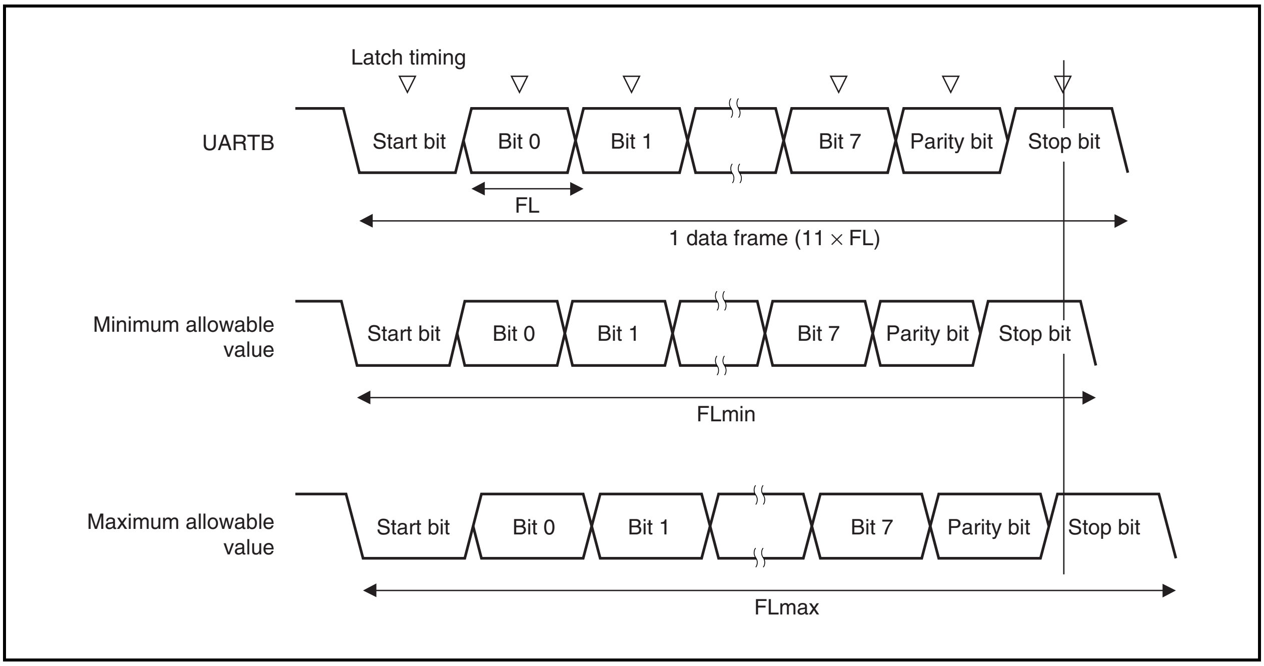
Remark fxx: Peripheral clock
k: Settings of UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits
ERR: Baud rate error [%]

(4) Allowable baud rate range during reception

The degree to which a discrepancy from the transmission destination's baud rate is allowed during reception is shown below.

Caution The equations described below should be used to set the baud rate error during reception so that it always is within the allowable error range.

Figure 15-10. Allowable Baud Rate Range During Reception



As shown in Figure 15-10, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the UBCTL2 register. If all data up to the final data (stop bit) is in time for this latch timing, the data can be received normally.
Applying this to 11-bit reception is, theoretically, as follows.

$FL = (Brate)^{-1}$

Brate: UARTB baud rate
k: UBCTL2 set value
FL: 1-bit data length
Latch timing margin: 2 clocks

$$\text{Minimum allowable value: } FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the maximum baud rate that can be received at the transfer destination is as follows.

$$\mathrm{BRmax} = (\mathrm{FLmin}/11)^{-1} = \frac{22k}{21k + 2}\,\mathrm{Brate}$$

Similarly, the maximum allowable value can be obtained as follows.

$$\frac{10}{11} \times \mathrm{FLmax} = 11 \times \mathrm{FL} - \frac{k + 2}{2 \times k} \times \mathrm{FL} = \frac{21k - 2}{2 \times k}\,\mathrm{FL}$$

$$\mathrm{FLmax} = \frac{21k - 2}{20k}\,\mathrm{FL} \times 11$$

Therefore, the minimum baud rate that can be received at the transfer destination is as follows.

$$\mathrm{BRmin} = (\mathrm{FLmax}/11)^{-1} = \frac{20k}{21k - 2}\,\mathrm{Brate}$$

The allowable baud rate error of UARTB and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

Table 15-7. Maximum and Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.33 %	–2.44
8	+3.53 %	–3.61
16	+4.14 %	–4.19
32	+4.45 %	–4.48
64	+4.61 %	–4.62
128	+4.68 %	–4.69
256	+4.72 %	–4.73
512	+4.74 %	–4.74
1024	+4.75 %	–4.75
2048	+4.76 %	–4.76
4096	+4.76 %	–4.76
8192	+4.76 %	–4.76
16384	+4.76 %	–4.76
32768	+4.76 %	–4.76
65535	+4.76 %	–4.76

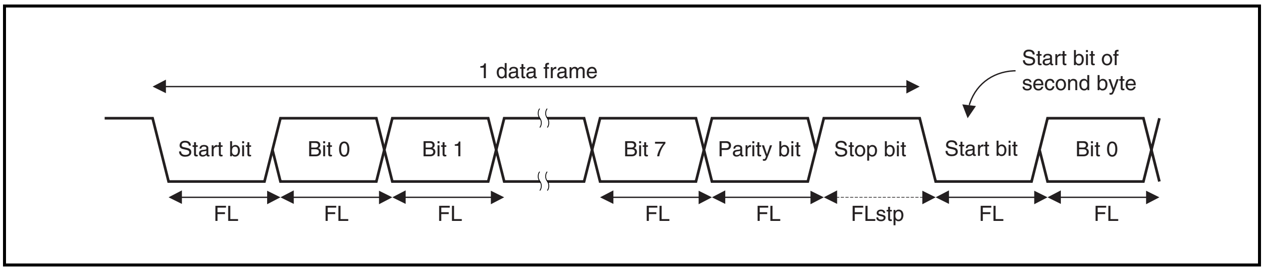
Remarks 1. The reception precision depends on the number of bits in one frame, the base clock frequency, and the division ratio (k). The higher the base clock frequency and the larger the division ratio (k), the higher the precision.

2. k: UBCTL2 set value

(5) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from a stop bit to the next start bit is extended two clocks longer than normal. However, on the reception side, the transfer result is not affected since the timing is initialized by the detection of the start bit.

Figure 15-11. Transfer Rate During Continuous Transmission



Representing the 1-bit data length by FL, the stop bit length by FLstp, and the base clock frequency by fxx/2 yields the following equation.

FLstp = FL + 2/(fxx/2)

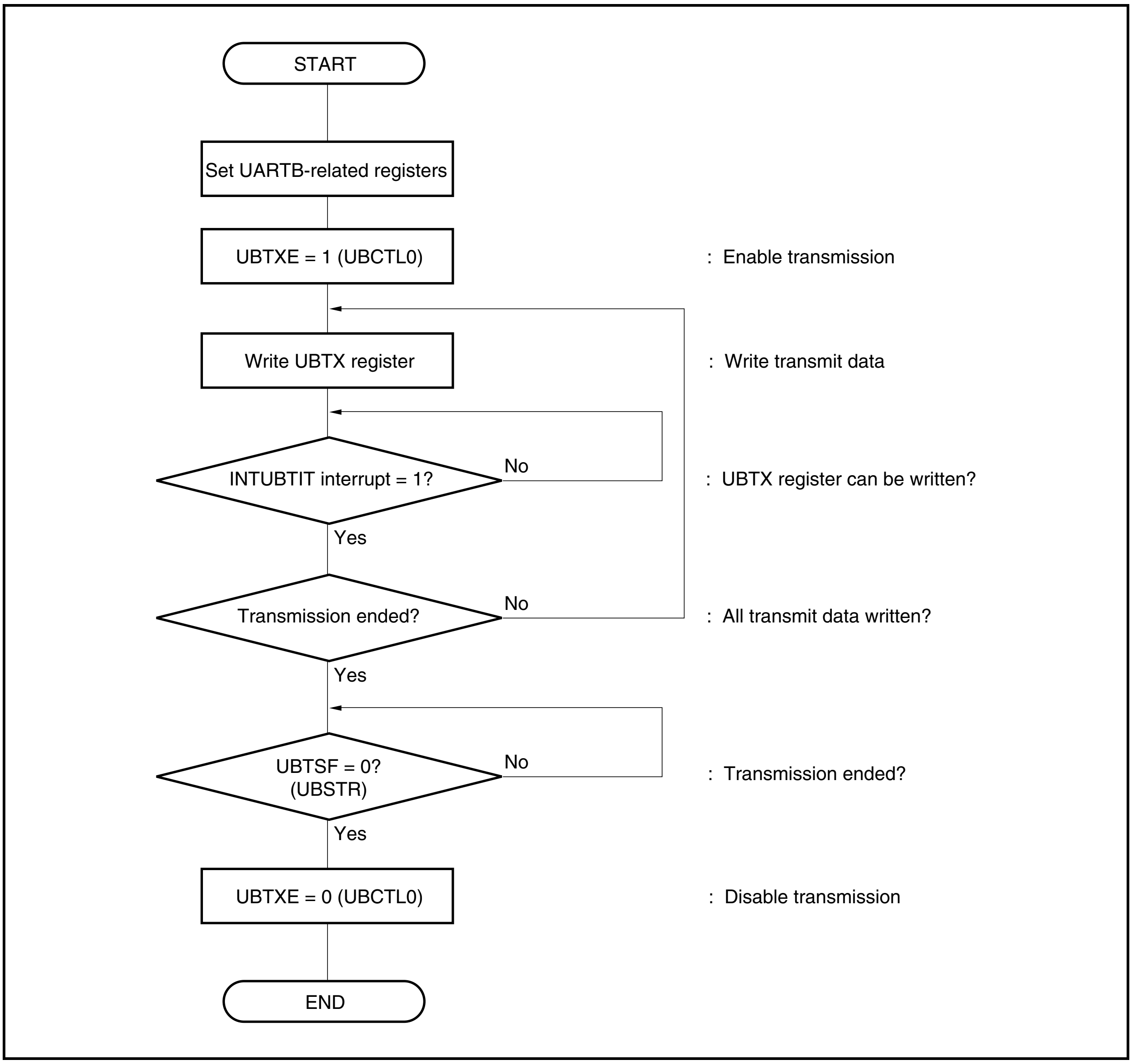
Therefore, the transfer rate during continuous transmission is as follows.

Transfer rate = 11 × FL + 2/(fxx/2)

15.9 Control Flow

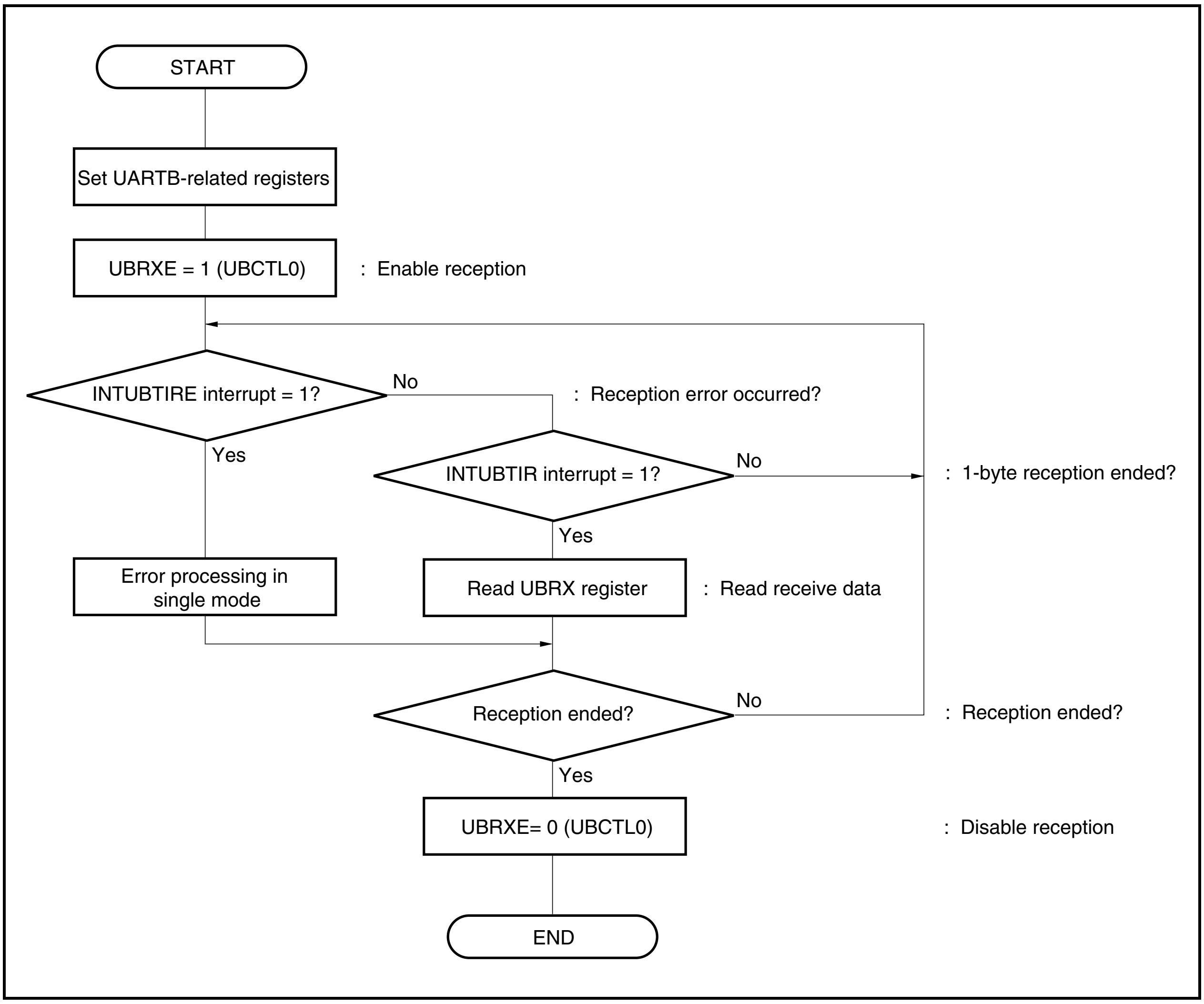
(1) Example of continuous transmission processing flow in single mode (CPU control)

Figure 15-12. Example of Continuous Transmission Processing Flow in Single Mode (CPU Control)



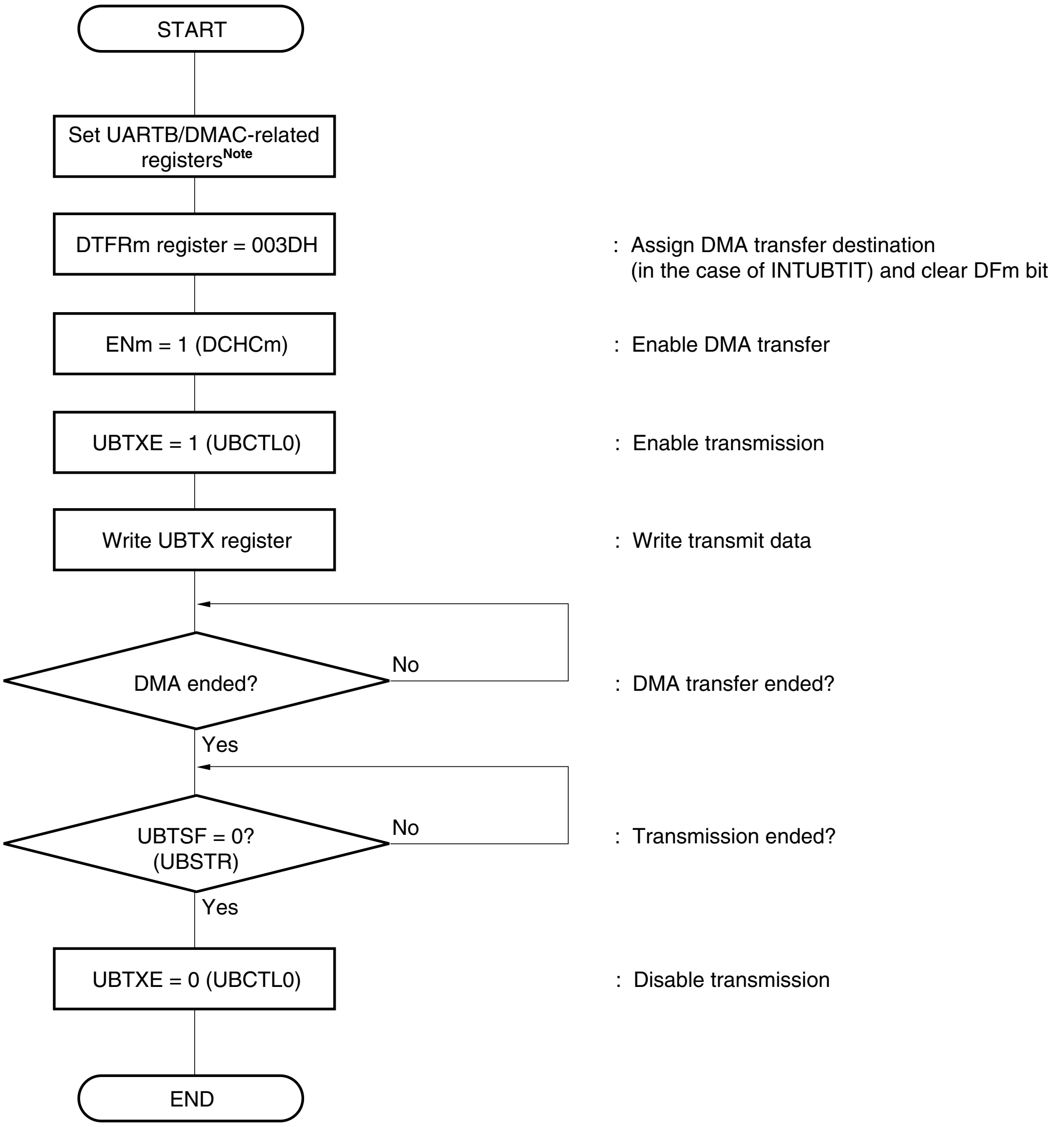
(2) Example of continuous reception processing flow in single mode (CPU control)

Figure 15-13. Example of Continuous Reception Processing Flow in Single Mode (CPU Control)



(3) Example of continuous transmission processing flow in single mode (DMA control)

Figure 15-14. Example of Continuous Transmission Processing Flow in Single Mode (DMA Control)

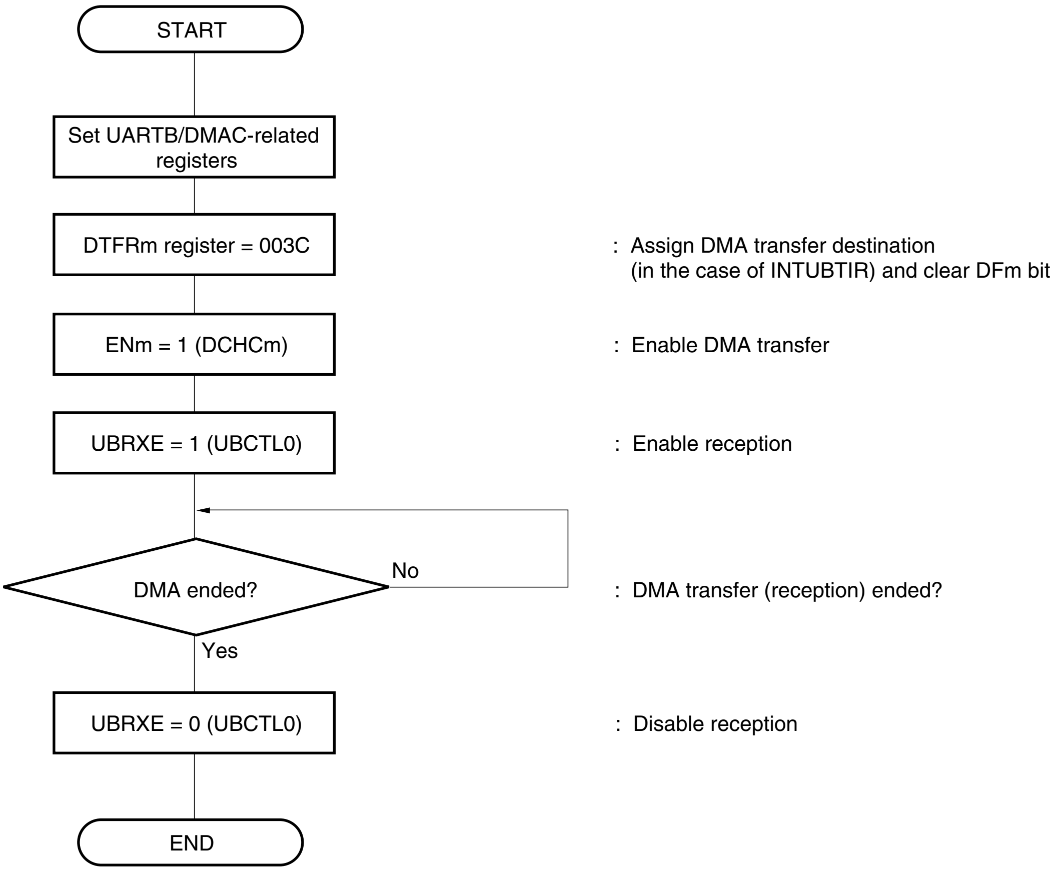


Note In this control flow example, transmission of the first byte of the data is executed by a CPU write operation. Exercise care in setting the number of data for DMA transfer (DTCRm register) and the source address (DSARm, DSARmH, and DSARmL registers).

Remark m = 0 to 6

(4) Example of continuous reception processing flow in single mode (DMA control)

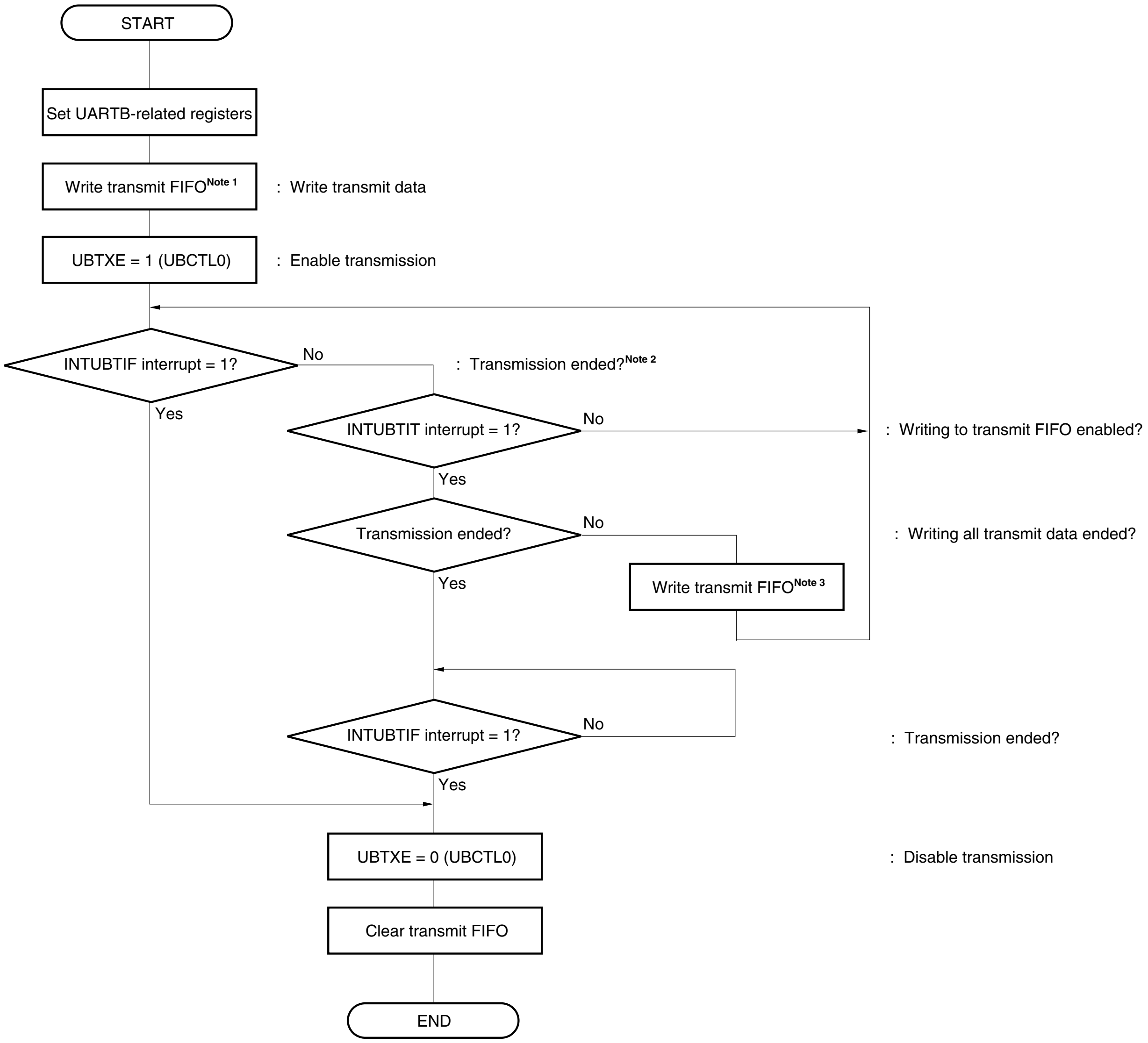
Figure 15-15. Example of Continuous Reception Processing Flow in Single Mode (DMA Control)



Remark m = 0 to 6

(5) Example of continuous transmission processing flow in FIFO mode (CPU control)

Figure 15-16. Example of Continuous Transmission Processing Flow in FIFO Mode (CPU Control)



Notes 1. Write more transmit data than the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to transmit FIFO.

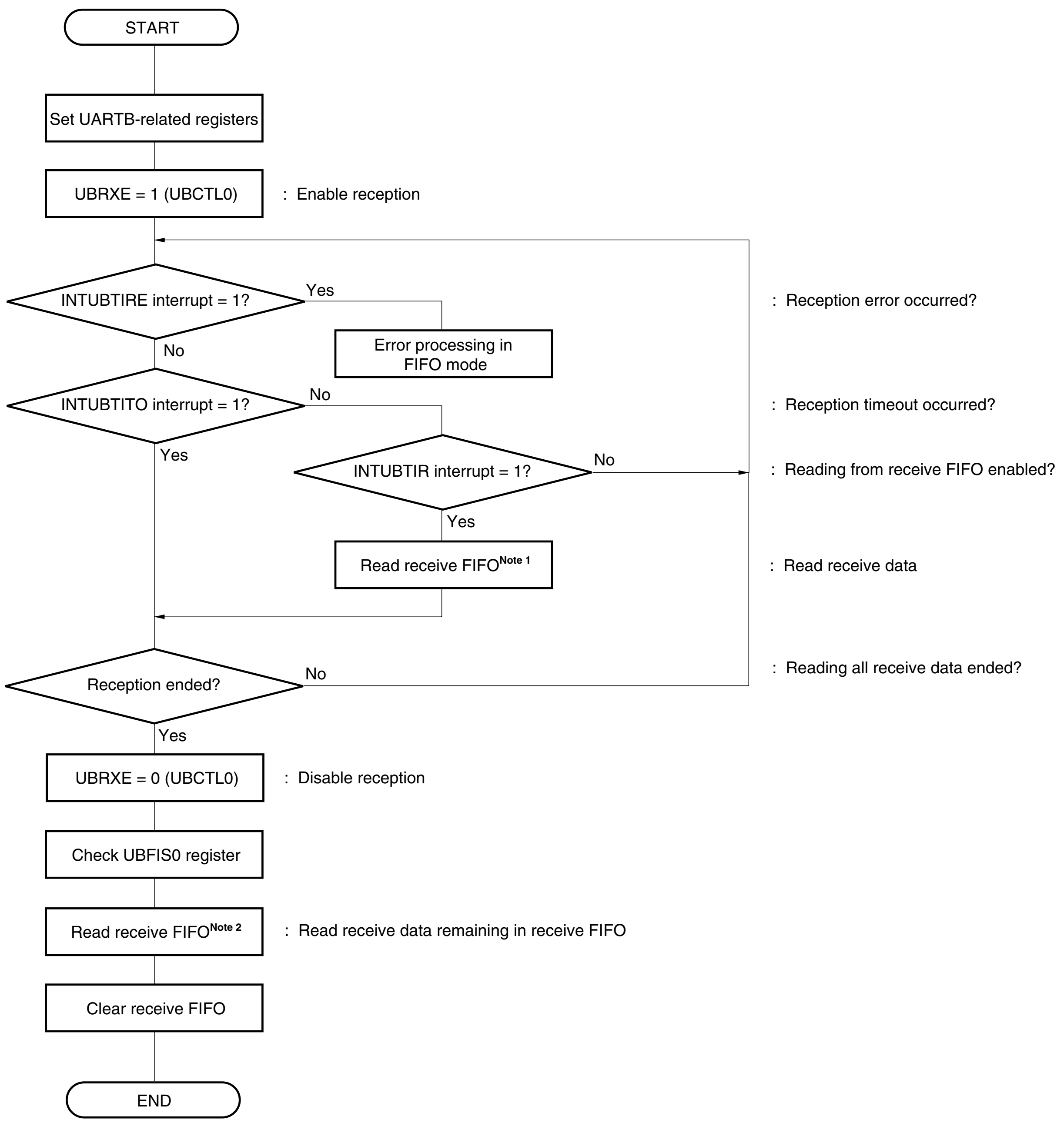
2. This is the case where transmission is ended (transmit FIFO and the transmit shift register become empty) before the next transmit data is written. To continue data transmission, clear the INTUBTIF and INTUBTIT signals and write the next data to transmit FIFO.

3. In the pending mode (UBFIC0.UBITM bit = 0), write as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits of to transmit FIFO. In the pointer mode (UBITM bit = 1), reference the UBFIS1.UBTB4 to UBFIS1.UBTB0 bits and write as many data as the number of empty bytes in transmit FIFO to transmit FIFO.

Write 16-byte data to fully use the 8-bit × 16-stage FIFO function.

(6) Example of continuous reception processing in FIFO mode (CPU control)

Figure 15-17. Example of Continuous Reception Processing in FIFO Mode (CPU Control)

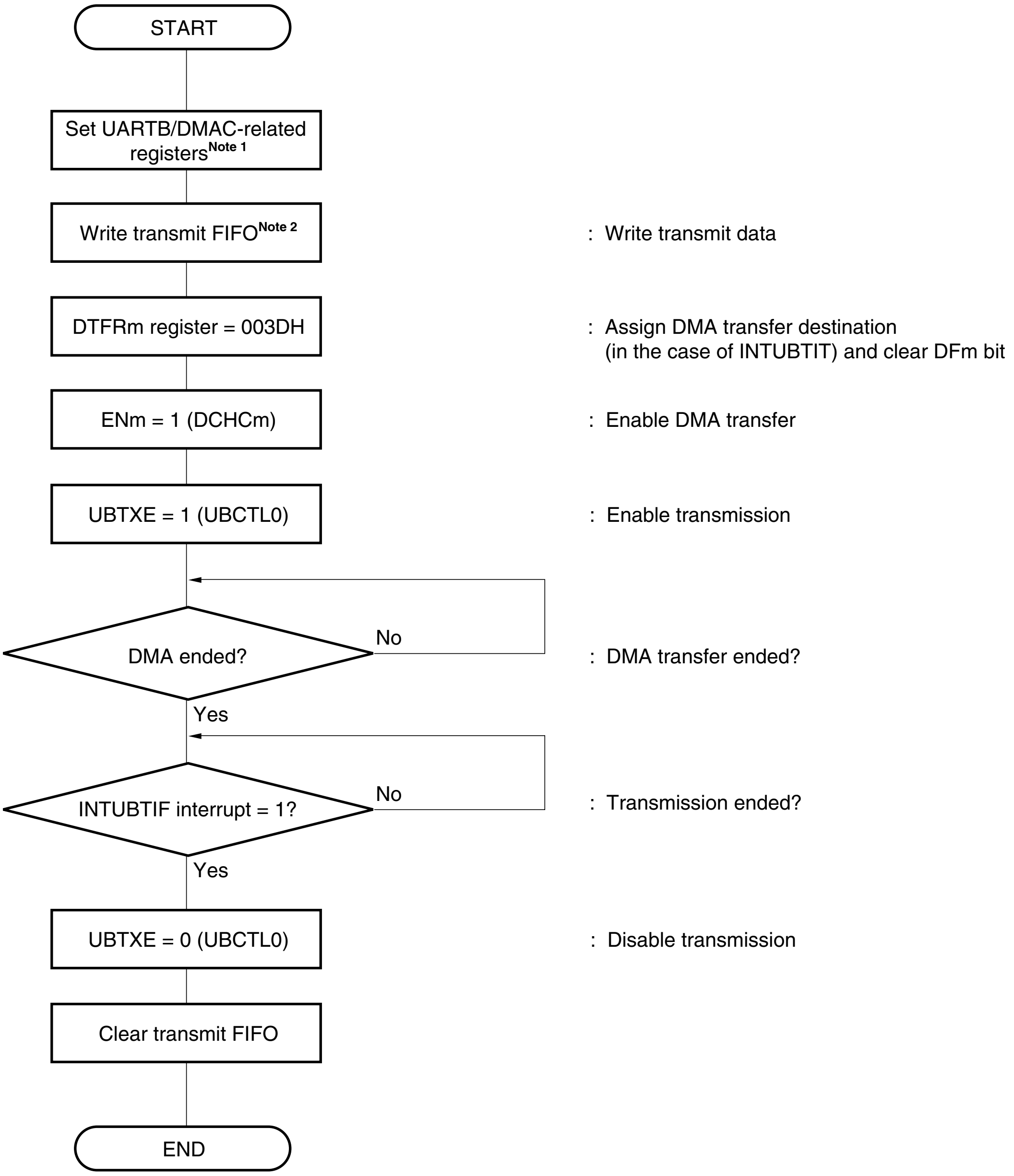


Notes 1. Read as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits from receive FIFO in the pending mode (UBFIC0.UBIRM bit = 0). In the pointer mode (UBIRM bit = 1), reference the UBFIS0.UBRB4 to UBFIS0.UBRB0 bits and read as many data as the number of bytes stored in receive FIFO from receive FIFO.

2. Read as many data (remaining receive data less than the number set as the trigger) as the number of bytes stored in receive FIFO from receive FIFO by referencing the UBFIS0.UBRB4 to UBFIS0.UBRB0 bits.

(7) Example of continuous transmission (pending mode) processing in FIFO mode (DMA control)

Figure 15-18. Example of Continuous Transmission (Pending Mode) Processing in FIFO Mode (DMA Control)



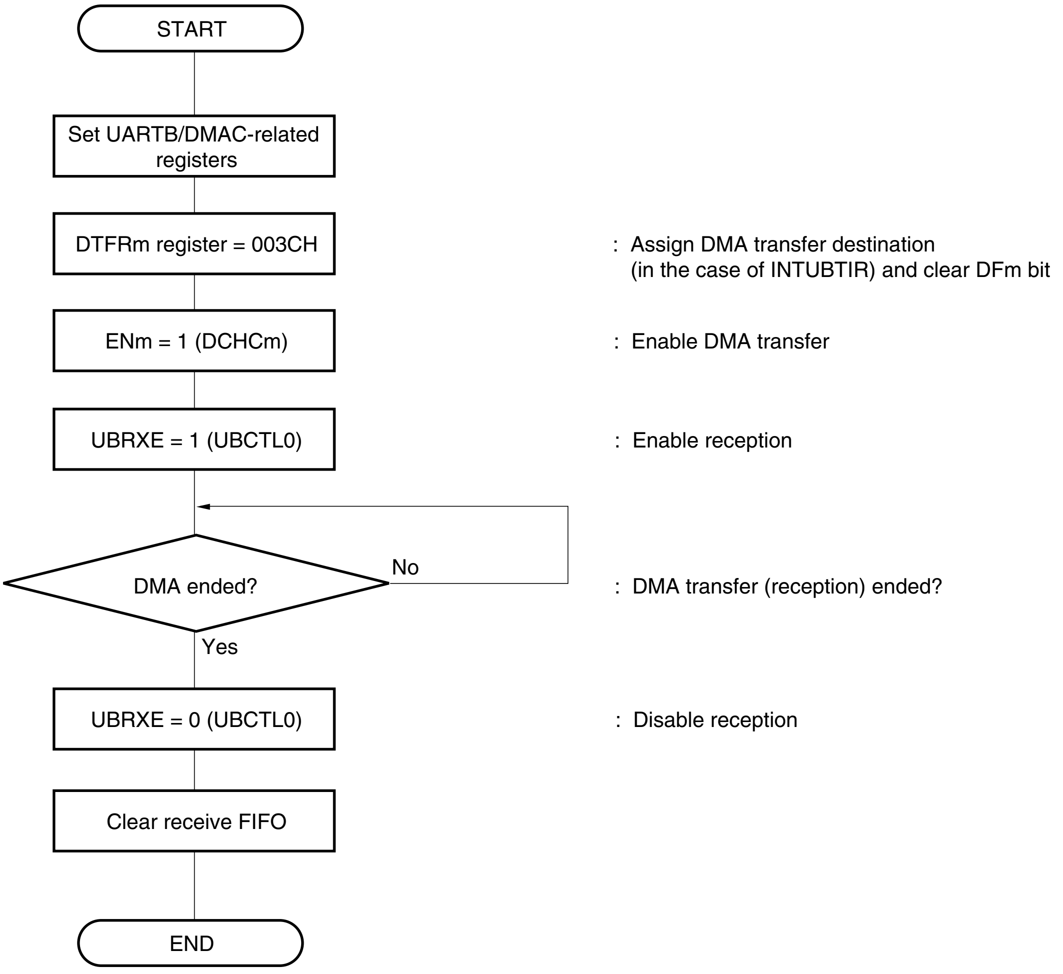
Notes 1. In this control flow example, transmission of the data described in **Note 2** is executed by a CPU write operation. Exercise care in setting the number of data for DMA transfer (DTCRm register) and the source address (DSARm, DSARmH, and DSARmL registers).

2. Write as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits (= 1 byte) to transmit FIFO.

Remark m = 0 to 6

(8) Example of continuous reception (pending mode) processing flow in FIFO mode (DMA control)

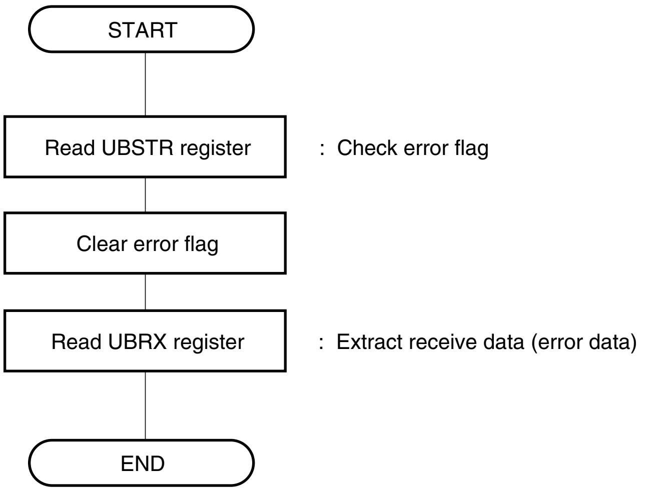
Figure 15-19. Example of Continuous Reception (Pending Mode) Processing Flow in FIFO Mode (DMA Control)



Remark m = 0 to 6

(9) Example of reception error processing in single mode

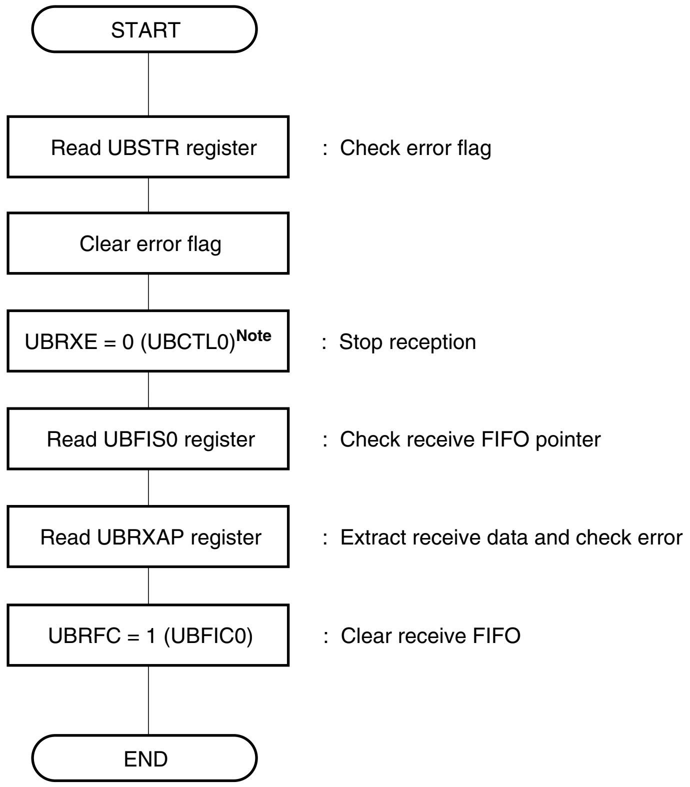
Figure 15-20. Example of Reception Error Processing Flow in Single Mode



Caution Reception can be continued by completing this control flow before reception of the next data is ended. If the next data is received before this control flow is ended, a reception error interrupt request signal (INTUBTIRE) may occur even if the data has been received correctly.

(10) Example of reception error processing flow in FIFO mode (1)

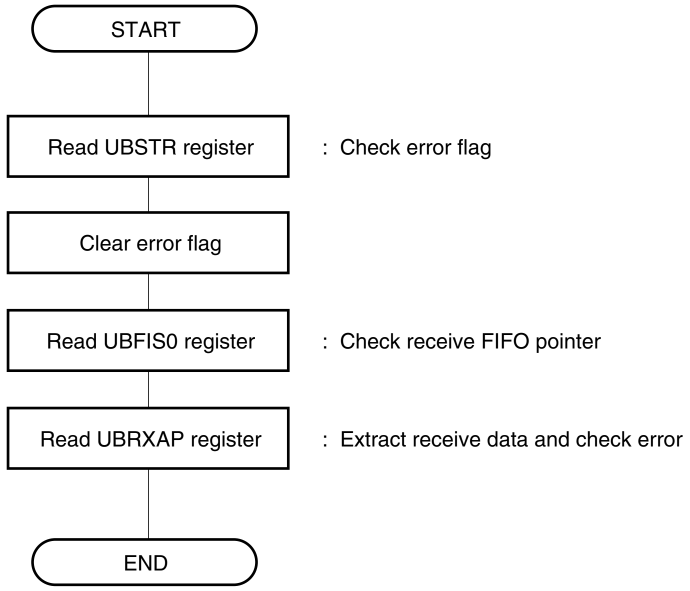
Figure 15-21. Example of Reception Error Processing Flow in FIFO Mode (1)



Note If the error flag is cleared when UBRXE bit = 0, the UBCTL0 register does not have to be set.

(11) Example of reception error processing flow in FIFO mode (2)

Figure 15-22. Example of Reception Error Processing Flow in FIFO Mode (2)



Caution Reception can be continued by completing this control flow before reception of the next data is ended. Extract the receive data and check if a reception error has occurred before receive FIFO becomes empty. Note that this control flow is valid only when a parity error or a framing error occurs. If an overflow error occurs, receive FIFO must be cleared (UBFIC0.UBRFC bit = 1).

If the next data is received before this control flow is ended, a reception error interrupt request signal (INTUBTIRE) may occur even if the data has been received correctly.

15.10 Cautions

Cautions concerning UARTB are shown below.

(1) When supply clock to UARTB is stopped

When the supply of clocks to UARTB is stopped (for example, IDLE and STOP modes), operation stops with each register retaining the value it had immediately before the supply of clocks was stopped. The TXDB pin output also holds and outputs the value it had immediately before the supply of clocks was stopped. However, operation is not guaranteed after the supply of clocks is restarted. Therefore, after the supply of clocks is restarted, the circuits should be initialized by setting the UBPWR bit = 0, UBRXE bit = 0, and UBTXE bit = 0.

(2) Caution on setting UBCTL0 register

- When using UARTB, set the external pins related to the UARTB function to the alternate function and set the UBCTL2 register. Then set the UBCTL0.UBPWR bit to 1 before setting the other bits.
- Be sure to input a high level to the RXDB pin when setting the external pins related to the UARTB function to the alternate function. If a low level is input, it is judged that a falling edge is input after the UBCTL0.UBRXE bit has been set to 1, and reception may be started.

(3) Caution on setting UBFIC2 register

Be sure to clear the UBCTL0.UBTXE bit (to disable transmission) and UBCTL0.UBRXE bit (to disable reception) to 0 before writing data to the UBFIC2 register. If data is written to the UBFIC2 register with the UBTXE or UBRXE bit set to 1, the operation is not guaranteed.

(4) Transmission interrupt request signal

In the single mode, the transmission enable interrupt request signal (INTUBTIT) occurs when the UBTX register becomes empty (when 1 byte of data is transferred from the UBTX register to the transmit shift register). In the FIFO mode, the FIFO transmission end interrupt request signal (INTUBTIF) occurs when data is no longer in transmit FIFO and the transmit shift register (when the FIFO and register are empty). However, the INTUBTIT signal or INTUBTIF signal does not occur if the transmit data register becomes empty due to $\overline{\text{RESET}}$ input.

(5) Initialization during continuous transmission in single mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing. If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

(6) Initialization during continuous transmission (pending mode) in FIFO mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by checking the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

To write transmit data to transmit FIFO by DMA control, set the number of transmit data specified as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 1 byte; otherwise the operation will not be guaranteed.

(7) Initialization during continuous transmission (pointer mode) in FIFO mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by checking the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

(8) Receive operation in FIFO mode (pointer mode specified)

If the pointer mode is specified in the FIFO mode and if as many data as the number of bytes stored in receive FIFO are read by referencing the UBFIS0 register, no data may be stored in receive FIFO (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000) even though the reception end interrupt request signal (INTUBTIR) has occurred. In this case, do not read data from receive FIFO. Be sure to read data from receive FIFO after confirming that the number of bytes stored in receive FIFO = 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).

CHAPTER 16 CLOCKED SERIAL INTERFACE F (CSIF)

16.1 Features

- Transfer rate: 6.25 Mbps (using internal clock)
- Master mode and slave mode selectable
- Interrupt request signals: 3
  - Reception end interrupt request signal (INTCFnR): This signal is generated when reception is enabled and receive data is transferred from the shift register to the CSIFn receive data register (CFnRX) after completion of a serial transfer.
  - Transmission enable interrupt request signal (INTCFnT): This signal is generated when transmission is enabled in the continuous transmission or continuous transmission/reception mode and transmission data is transferred from the CSIFn transmit data register (CFnTX) to the shift register.
  - Reception error interrupt request signal (INTCFnRE): This signal is generated if an overrun error occurs (CFnSTR.CFnOVE bit = 1) when reception is enabled in the continuous transfer mode.
- Serial clock and data phase switchable
- 3-wire serial interface, transfer data length selectable in 1-bit units between 8 and 16 bits
- Transfer data MSB-first/LSB-first switchable
- 3-wire transfer SOFn: Serial data output
  SIFn: Serial data input
  $\overline{\text{SCKFn}}$: Serial clock I/O
  Transmission mode, reception mode, and transmission/reception mode specifiable
- Double buffer for both transmission and reception
- Overrun error detection

Remark n = 0 to 2

16.2 Configuration

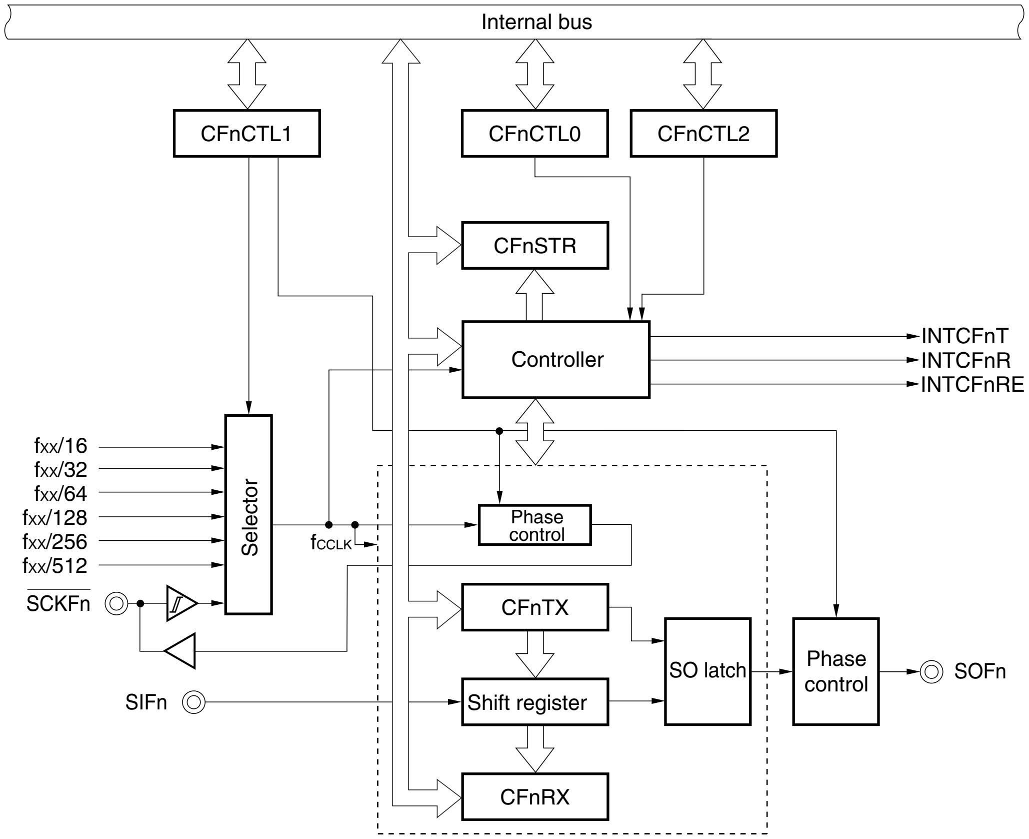
CSIFn includes the following hardware.

Table 16-1. Configuration of CSIFn

Item	Configuration
Registers	CSIFn receive data register (CFnRX) CSIFn transmit data register (CFnTX)
Control registers	CSIFn control register 0 (CFnCTL0) CSIFn control register 1 (CFnCTL1) CSIFn control register 2 (CFnCTL2) CSIFn status register (CFnSTR)

The following shows the block diagram of CSIFn.

Figure 16-1. Block Diagram of CSIFn



Remarks 1. n = 0 to 2

2. fCCLK: Communication clock (6.25 MHz (max.))

16.2.1 Pin functions of each channel

The SIFn, SOFn, and $\overline{\text{SCKFn}}$ pins used by CSIF in the V850E/IG4-H and V850E/IH4-H are used alternately for other functions as shown in Table 16-2. To use these pins for CSIF, set up the related registers as described in **Table 4-16 Settings When Pins Are Used for Alternate Functions**.

Table 16-2. Pins Used by CSIF

Channel	Pin No.		Port	CSIF Reception Input	CSIF Transmission Output	CSIF Clock I/O	Other Functions
	IG4-H	IH4-H					
	GC	GF					
CSIF0	46	96	P40	SIF0	–	–	RXDA0/DDI/TOA00
	47	97	P41	–	SOF0	–	TXDA0
	48	98	P42	–	–	$\overline{\text{SCKF0}}$	DCK/TOA10
CSIF1	56	108	P32	SIF1	–	–	RXDA2/$\overline{\text{CS1}}$
	57	109	P33	–	SOF1	–	TXDA2
	58	110	P34	–	–	$\overline{\text{SCKF1}}$	INTP11/$\overline{\text{CS0}}$
CSIF2	59	111	P35	SIF2	–	–	RXDB
	60	112	P36	–	SOF2	–	TXDB
	61	113	P37	–	–	$\overline{\text{SCKF2}}$	INTP12/ASTB

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

16.3 Mode Switching Between CSIF and Other Serial Interface

16.3.1 Mode switching between CSIF0 and UARTA0

In the V850E/IG4-H and V850E/IH4-H, CSIF0 and UARTA0 share a pin, and these functions cannot be used at the same time. To use the pin for the CSIF0 function, set up the PMC4, PFC4, and PFCE4 registers in advance.

Switching the operation mode between CSIF0 and UARTA0, the serial interfaces, is described below.

Caution The operations related to transmission and reception of CSIF0 or UARTA0 are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-2. Operation Mode Switch Settings of CSIF0 and UARTA0

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	0	0	PMC44	PMC43	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	PFC44	PFC43	0	PFC41	PFC40

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	0	0	0	0	0	PFCE42	0	PFCE40

PMC42	PFCE42	Operation mode
0	×	Port I/O mode
1	0	$\overline{\text{SCKF0}}$

PMC4n	PFC4n	Operation mode
0	×	Port I/O mode
1	0	CSIF0 mode
1	1	UARTA0 mode

Remarks 1. n = 0, 1

2. × = 0 or 1

16.3.2 Mode switching between CSIF1 and UARTA2

In the V850E/IG4-H and V850E/IH4-H, CSIF1 and UARTA2 share a pin, and these functions cannot be used at the same time. To use the pin for the CSIF1 function, set up the PMC3, PFC3, and PFCE3 registers in advance.

Switching the operation mode between CSIF1 and UARTA2, the serial interfaces, is described below.

Caution The operations related to transmission and reception of CSIF1 or UARTA2 are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-3. Operation Mode Switch Settings of CSIF1 and UARTA2

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	0	0	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC34	PFC34	Operation mode
0	×	Port I/O mode
1	0	$\overline{\text{SCKF1}}$ I/O

PMC3n	PFC3n	Operation mode
0	×	Port I/O mode
1	0	CSIF1 mode
1	1	UARTA2 mode

Remarks 1. n = 2, 3

2. × = 0 or 1

16.3.3 Mode switching between CSIF2 and UARTB

In the V850E/IG4-H and V850E/IH4-H, CSIF2 and UARTB share a pin, and these functions cannot be used at the same time. To use the pin for the CSIF2 function, set up the PMC3, PFC3, and PFCE3 registers in advance.

Switching the operation mode between CSIF2 and UARTB, the serial interfaces, is described below.

Caution The operations related to transmission and reception of CSIF2 or UARTB are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-4. Operation Mode Switch Settings of CSIF2 and UARTB

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	0	0	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC37	PFC37	Operation mode
0	×	I/O port
1	0	$\overline{\text{SCKF2}}$

PMC3n	PFC3n	Operation mode
0	×	I/O port
1	0	CSIF2 mode
1	1	UARTB mode

Remarks 1. n = 5, 6

2. × = 0 or 1

16.4 Control Registers

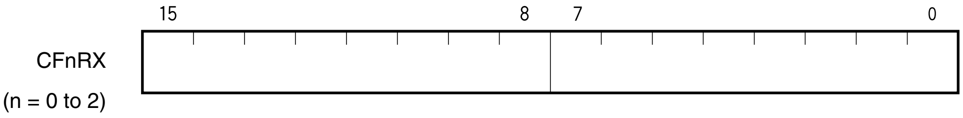
The registers that control CSIFn are shown below.

- CSIFn reception data register (CFnRX)
- CSIFn transmission data register (CFnTX)
- CSIFn control register 0 (CFnCTL0)
- CSIFn control register 1 (CFnCTL1)
- CSIFn control register 2 (CFnCTL2)
- CSIFn status register (CFnSTR)

(1) CSIFn receive data register (CFnRX)

The CFnRX register is a 16-bit buffer register that holds receive data.
This register is read-only, in 16-bit units.
The receive operation is started by reading the CFnRX register during the reception mode.
If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CFnRXL register.
Reset sets this register to 0000H.
In addition to reset, the CFnRX register can be initialized by clearing (to 0) the CFnCTL0.CFnPWR bit.

After reset: 0000H R Address: CF0RX FFFFFD04H, CF0RXL FFFFFD04H,
CF1RX FFFFFD14H, CF1RXL FFFFFD14H,
CF2RX FFFFFD24H, CF2RXL FFFFFD24H



(2) CSIFn transmit data register (CFnTX)

The CFnTX register is a 16-bit buffer register used to write the CSIFn transfer data.

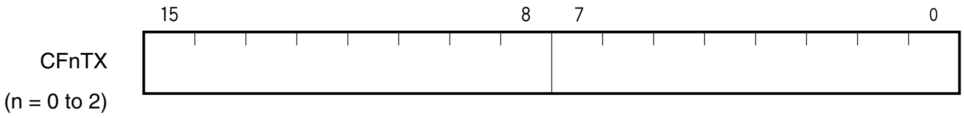
This register can be read or written in 16-bit units.

The transmit operation is started by writing data to the CFnTX register during the transmission mode.

If the transfer data length is 8 bits, the lower 8 bits of this register can be read or written in 8-bit units as the CFnTXL register.

Reset sets this register to 0000H.

After reset: 0000H R/W Address: CF0TX FFFFFD06H, CF0TXL FFFFFD06H,
CF1TX FFFFFD16H, CF1TXL FFFFFD16H,
CF2TX FFFFFD26H, CF2TXL FFFFFD26H



Remark The communication start conditions are shown below.

Transmission mode (CFnTXE bit = 1, CFnRXE bit = 0):	Write to CFnTX register
Transmission/reception mode (CFnTXE bit = 1, CFnRXE bit = 1):	Write to CFnTX register
Reception mode (CFnTXE bit = 0, CFnRXE bit = 1):	Read from CFnRX register

(3) CSIFn control register 0 (CFnCTL0)

CFnCTL0 is a register that controls the CSIFn serial transfer operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 01H.

(1/2)

After reset: 01H R/W Address: CF0CTL0 FFFFFD00H, CF1CTL0 FFFFFD10H,
CF2CTL0 FFFFFD20H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CFnCTL0 (n = 0 to 2)	CFnPWR	CFnTXE[Note]	CFnRXE[Note]	CFnDIR[Note]	0	0	CFnTMS[Note]	CFnSCE

CFnPWR	Specification of CSIFn operation disable/enable
0	Disable CSIFn operation and reset the CFnSTR register
1	Enable CSIFn operation
• The CFnPWR bit controls the CSIFn operation and resets the internal circuit.	

CFnTXE[Note]	Specification of transmit operation disable/enable
0	Disable transmit operation
1	Enable transmit operation
• The SOFn output is low level when the CFnTXE bit is 0.	

CFnRXE[Note]	Specification of receive operation disable/enable
0	Disable receive operation
1	Enable receive operation
• When the CFnRXE bit is 0, no reception end interrupt is output even when the prescribed data is transferred in order to disable the receive operation, and the receive data (CFnRX register) is not updated.	

Note These bits can only be rewritten when the CFnPWR bit = 0. However, CFnPWR bit = 1 can also be set at the same time as rewriting these bits.

Caution Be sure to set bits 3 and 2 to “0”.

(2/2)

CFnDIR[Note 1]	Specification of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first

<table>
<tr><th>CFnTMS[Note 1]</th><th>Transfer mode specification</th></tr>
<tr><td>0</td><td>Single transfer mode</td></tr>
<tr><td>1</td><td>Continuous transfer mode</td></tr>
<tr><td colspan="2">• When using single transmission or transmission/reception mode with communication type 2 or 4 (CFnCTL1.CFnDAP bit = 1), write the transfer data to the CFnTX register after checking that the CFnSTR.CFnTSF bit is 0.
• When using DMA, use the continuous transfer mode.</td></tr>
</table>

<table>
<tr><th>CFnSCE</th><th>Specification of start transfer disable/enable</th></tr>
<tr><td>0</td><td>Communication start trigger invalid</td></tr>
<tr><td>1</td><td>Communication start trigger valid</td></tr>
<tr><td colspan="2">• In master mode
This bit enables or disables the communication start trigger.
(a) In single reception mode
Set the CFnSCE bit to 0 before reading the receive data (CFnRX register)[Note 2].
(b) In continuous reception mode
Set the CFnSCE bit to 0 one communication clock before reception of the last data is ended[Note 3].
• In slave mode
This bit enables or disables the communication start trigger.
(a) In single reception mode or continuous reception mode
Set the CFnSCE bit to 1[Note 4].
• In single transmission or transmission/reception mode, or continuous transmission or transmission/reception mode
The function of the CFnSCE bit is invalid. It is recommended to set this bit to 1.</td></tr>
</table>

Notes **1.** These bits can only be rewritten when the CFnPWR bit = 0. However, the CFnPWR bit can be set to 1 at the same time as these bits are rewritten.

2. If the CFnSCE bit is read while it is 1, the next communication operation is started.

3. The CFnSCE bit is not set to 0 one communication clock before the end of the last data reception, the next communication operation is automatically started.

To start communication operation again after reading the last data, set the CFnSCE bit to 1 and perform a dummy read of the CFnRX register.

4. To start the reception, a dummy read is necessary.

(a) How to use CFnSCE bit

(i) In single reception mode

<1> When the reception of the last data is ended with INTCFnR interrupt servicing, clear the CFnSCE bit to 0, and then read the CFnRX register.

<2> When the reception is disabled after the reception of the last data has been ended, check that the CFnSTR.CFnTSF bit is 0, and then clear the CFnPWR and CFnRXE bits to 0. To continue reception, set the CFnSCE bit to 1 and start the next receive operation by performing a dummy read of the CFnRX register.

(ii) In continuous reception mode

<1> Clear the CFnSCE bit to 0 during reception of the last data with INTCFnR interrupt servicing by the reception before the last reception, and then read the CFnRX register.

<2> After receiving the INTCFnR signal of the last reception, read the last data from the CFnRX register.

<3> When the reception is disabled after the reception of the last data has been ended, check that the CFnSTR.CFnTSF bit is 0, and then clear the CFnPWR and CFnRXE bits to 0. To continue reception, set the CFnSCE bit to 1 and start the next receive operation by performing a dummy read of the CFnRX register.

Caution In continuous reception mode, the serial clock is not stopped until the reception executed when the CFnSCE bit is cleared to 0 is ended after the reception is started by a dummy read.

(4) CSIFn control register 1 (CFnCTL1)

CFnCTL1 is an 8-bit register that controls the CSIFn serial transfer operation.

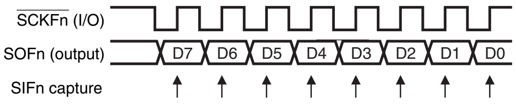
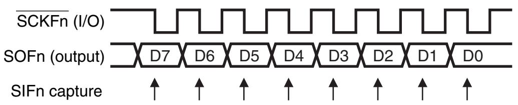
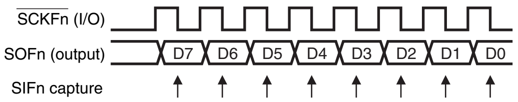
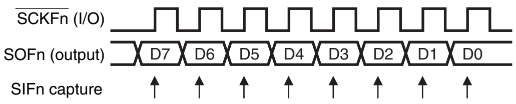
This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution The CFnCTL1 register can be rewritten only when the CFnCTL0.CFnPWR bit = 0.

After reset: 00H R/W Address: CF0CTL1 FFFFFD01H, CF1CTL1 FFFFFD11H,
CF2CTL1 FFFFFD21H

	7	6	5	4	3	2	1	0
CFnCTL1 (n = 0 to 2)	0	0	0	CFnCKP	CFnDAP	CFnCKS2	CFnCKS1	CFnCKS0

	CFnCKP	CFnDAP	Specification of data transmission/ reception timing in relation to $\overline{\text{SCKFn}}$
Communication type 1	0	0	 $\overline{\text{SCKFn}}$ (I/O) SOFn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIFn capture
Communication type 2	0	1	 $\overline{\text{SCKFn}}$ (I/O) SOFn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIFn capture
Communication type 3	1	0	 $\overline{\text{SCKFn}}$ (I/O) SOFn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIFn capture
Communication type 4	1	1	 $\overline{\text{SCKFn}}$ (I/O) SOFn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIFn capture

CFnCKS2	CFnCKS1	CFnCKS0	Communication clock (f_{CCLK})	Mode
0	0	0	f_{XX}/16	Master mode
0	0	1	f_{XX}/32	Master mode
0	1	0	f_{XX}/64	Master mode
0	1	1	f_{XX}/128	Master mode
1	0	0	f_{XX}/256	Master mode
1	0	1	f_{XX}/512	Master mode
1	1	0	Setting prohibited	Master mode
1	1	1	External clock ($\overline{\text{SCKFn}}$)	Slave mode

Caution Set f_{CCLK} to 6.25 MHz or lower.

(5) CSIFn control register 2 (CFnCTL2)

CFnCTL2 is an 8-bit register that controls the number of CSIFn serial transfer bits.

This register can be read or written in 8-bit units.

Reset sets register to 00H.

Caution The CFnCTL2 register can be rewritten only when the CFnCTL0.CFnPWR bit = 0 or when both the CFnTXE and CFnRXE bits = 0.

After reset: 00H R/W Address: CF0CTL2 FFFFFD02H, CF1CTL2 FFFFFD12H,
CF2CTL2 FFFFFD22H

	7	6	5	4	3	2	1	0
CFnCTL2 (n = 0 to 2)	0	0	0	0	CFnCL3	CFnCL2	CFnCL1	CFnCL0

CFnCL3	CFnCL2	CFnCL1	CFnCL0	Serial register bit length
0	0	0	0	8 bits
0	0	0	1	9 bits
0	0	1	0	10 bits
0	0	1	1	11 bits
0	1	0	0	12 bits
0	1	0	1	13 bits
0	1	1	0	14 bits
0	1	1	1	15 bits
1	×	×	×	16 bits

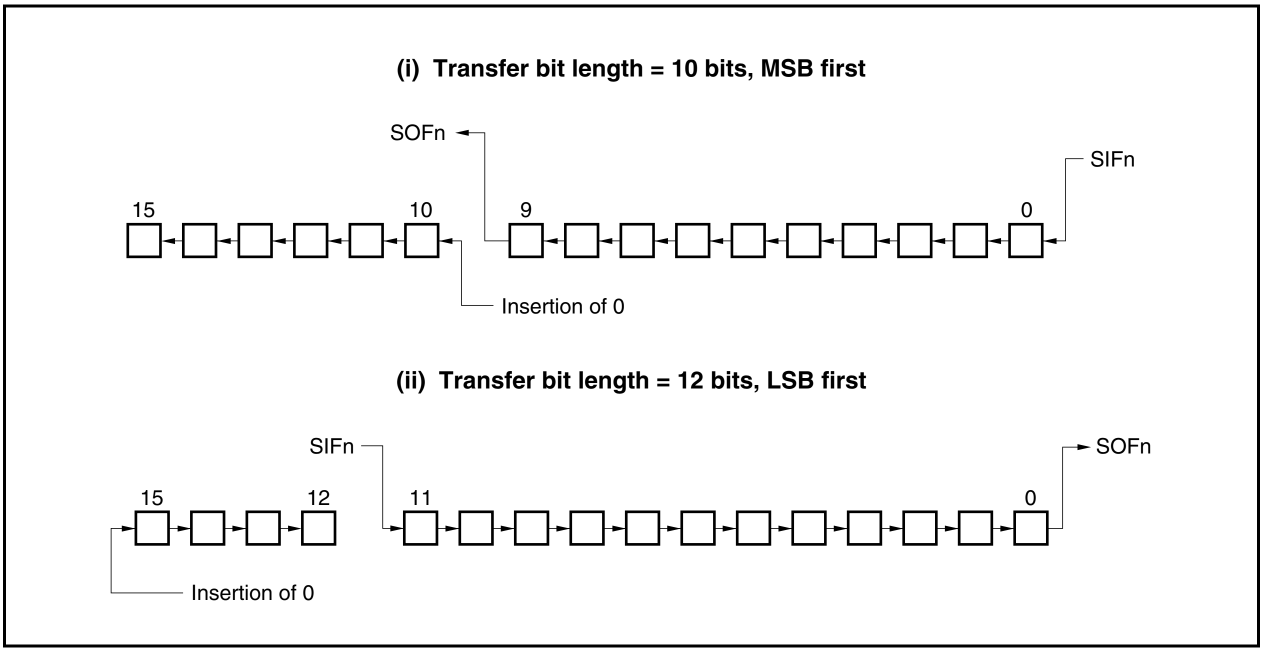
Remark If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CFnTX and CFnRX registers.

(a) Transfer data length change function

The CSIFn transfer data length can be set in 1-bit units between 8 and 16 bits using the CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits.

When the transfer bit length is set to a value other than 16 bits, set the data to the CFnTX or CFnRX register starting from the LSB, regardless of whether the transfer start bit is the MSB or LSB. Any data can be set for the higher bits that are not used, but the receive data becomes 0 following serial transfer.

Remark n = 0 to 2


(i) Transfer bit length = 10 bits, MSB first
SOFn
SIFn
15
10
9
0
Insertion of 0
(ii) Transfer bit length = 12 bits, LSB first
SIFn
SOFn
15
12
11
0
Insertion of 0

(6) CSIFn status register (CFnSTR)

CFnSTR is an 8-bit register that displays the CSIFn status.

This register can be read or written in 8-bit or 1-bit units, but the CFnTSF flag is read-only.

Reset sets this register to 00H.

In addition to reset, the CFnSTR register can be initialized by clearing (0) the CFnCTL0.CFnPWR bit.

After reset: 00H R/W Address: CF0STR FFFFFD03H, CF1STR FFFFFD13H,
CF2STR FFFFFD23H

	<7>	6	5	4	3	2	1	<0>
CFnSTR (n = 0 to 2)	CFnTSF	0	0	0	0	0	0	CFnOVE

CFnTSF	Communication status flag
0	Communication stopped
1	Communicating

- During transmission, this register is set when data is prepared in the CFnTX register, and during reception, it is set when a dummy read of the CFnRX register is performed.
  When transfer ends, this flag is cleared to 0 at the last edge of the clock.

CFnOVE	Overrun error flag
0	No overrun
1	Overrun

- An overrun error occurs when the next reception starts without performing a CPU read of the value of the CFnRX register, upon end of the receive operation.
  The CFnOVE flag displays the overrun error occurrence status in this case.
- The CFnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.

Caution In single transfer mode, writing to the CFnTX register with the CFnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.

For example, if the next data is written to the CFnTX register when DMA is started by generating the INTCFnR signal, the written data is not transferred because the CFnTSF bit is set to 1.

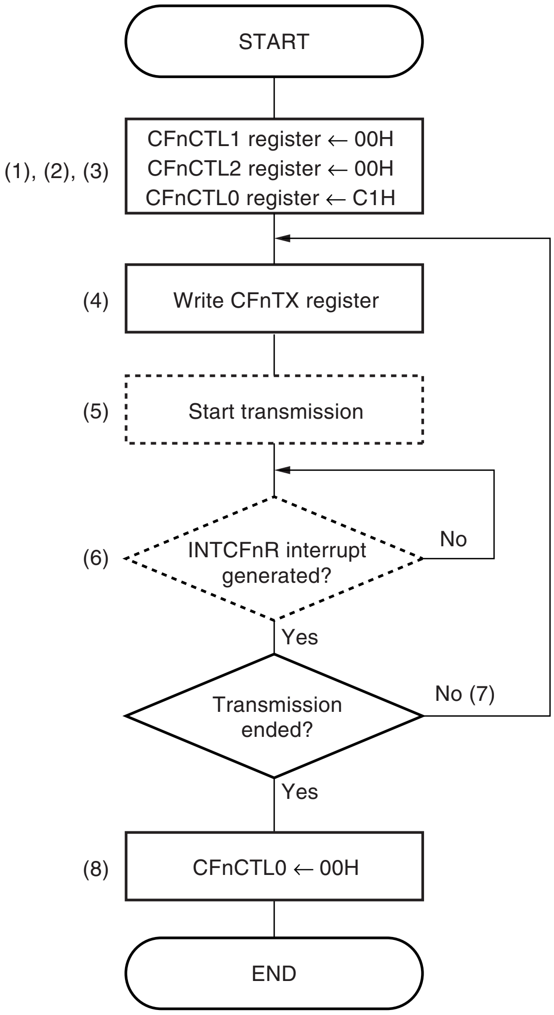
Use the continuous transfer mode, not the single transfer mode, for such applications.

16.5 Operation

16.5.1 Single transfer mode (master mode, transmission mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 000), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

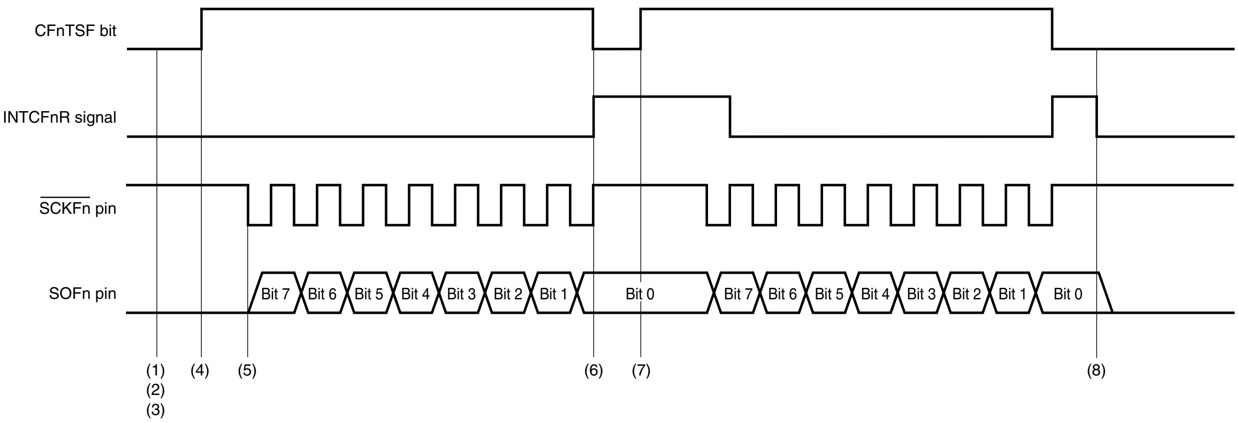


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CFnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by writing the transmit data to the CFnTX register, and transmission is started.

(5) When transmission is started, output the serial clock to the $\overline{\text{SCKFn}}$ pin, and output the transmit data from the SOFn pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CFnCTL2 register is completed, stop the serial clock output and transmit data output, generate the reception end interrupt request signal (INTCFnR) at the last edge of the serial clock, and clear the CFnTSF bit to 0.

(7) To continue transmission, start the next transmission by writing the transmit data to the CFnTX register again after the INTCFnR signal is generated.

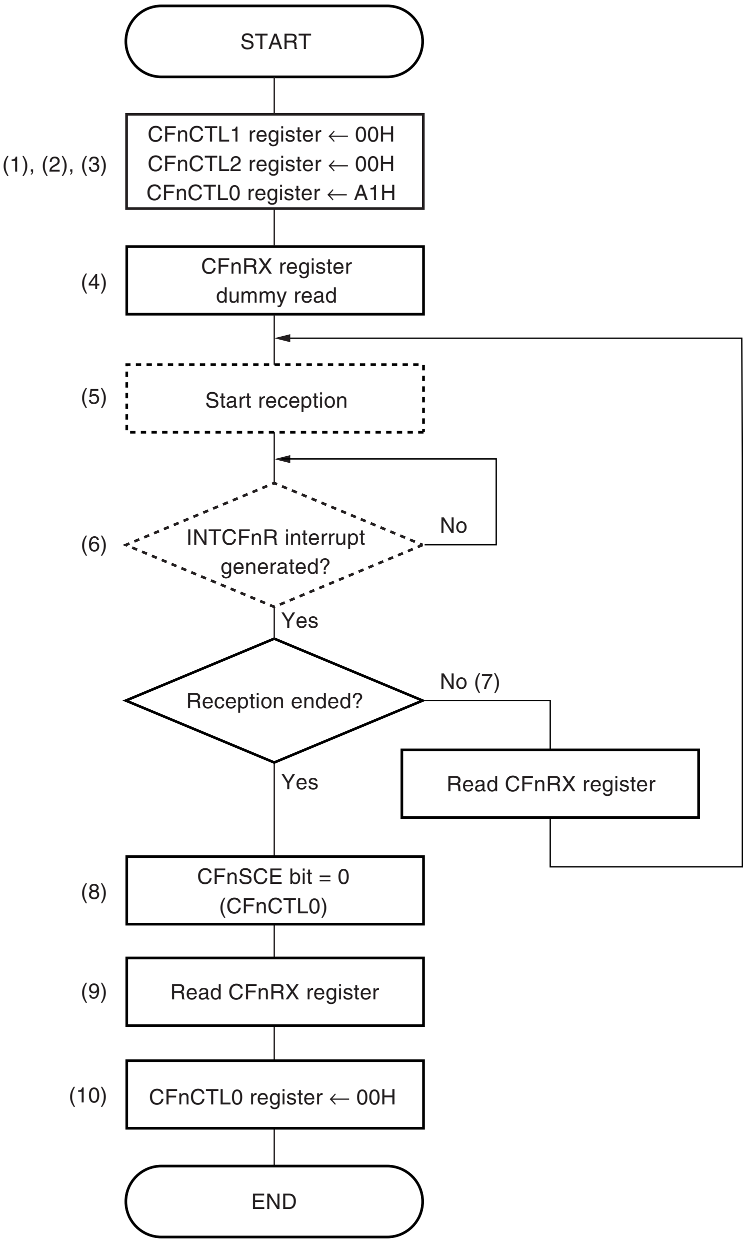
(8) To end transmission, write the CFnCTL0.CFnPWR bit = 0 and the CFnCTL0.CFnTXE bit = 0.

Remark n = 0 to 2

16.5.2 Single transfer mode (master mode, reception mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 000), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

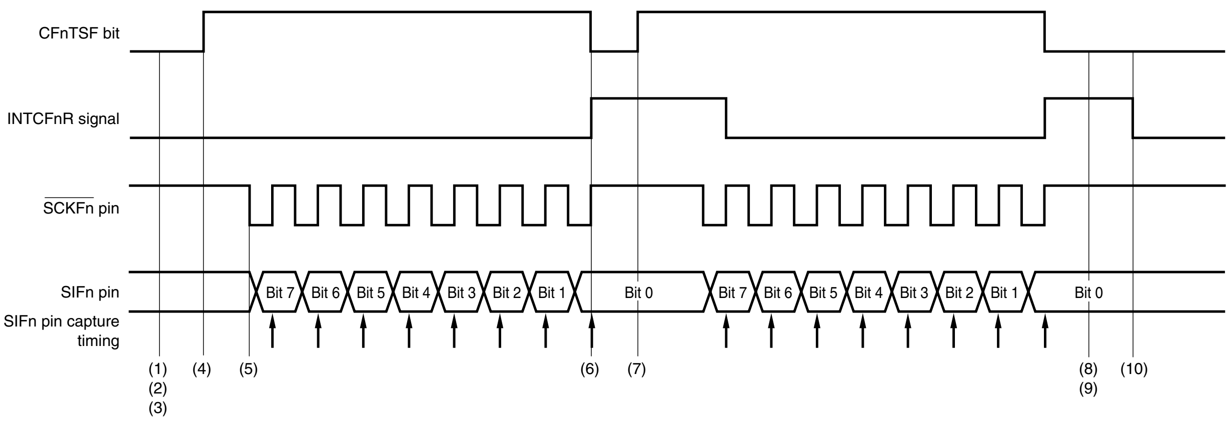


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CFnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by performing a dummy read of the CFnRX register, and reception is started.

(5) When reception is started, output the serial clock to the $\overline{\text{SCKFn}}$ pin, and capture the receive data of the SIFn pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CFnCTL2 register is completed, stop the serial clock output and data capturing, generate the reception end interrupt request signal (INTCFnR) at the last edge of the serial clock, and clear the CFnTSF bit to 0.

(7) To continue reception, read the CFnRX register with the CFnCTL0.CFnSCE bit = 1 remained after the INTCFnR signal is generated.

(8) To read the CFnRX register without starting the next reception, write the CFnSCE bit = 0.

(9) Read the CFnRX register.

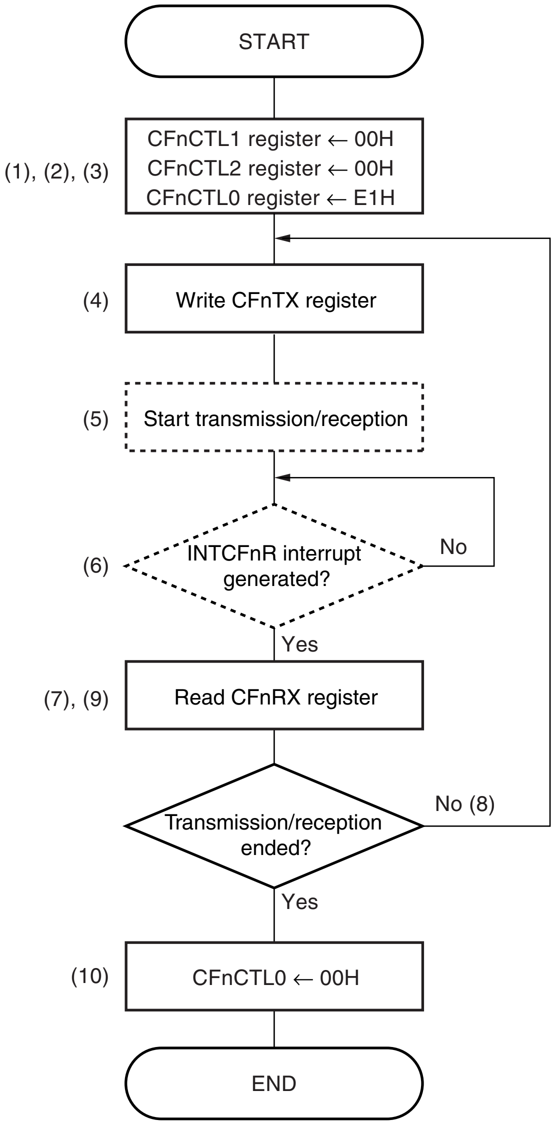
(10) To end reception, write the CFnCTL0.CFnPWR bit = 0 and the CFnCTL0.CFnRXE bit = 0.

Remark n = 0 to 2

16.5.3 Single transfer mode (master mode, transmission/reception mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 000), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

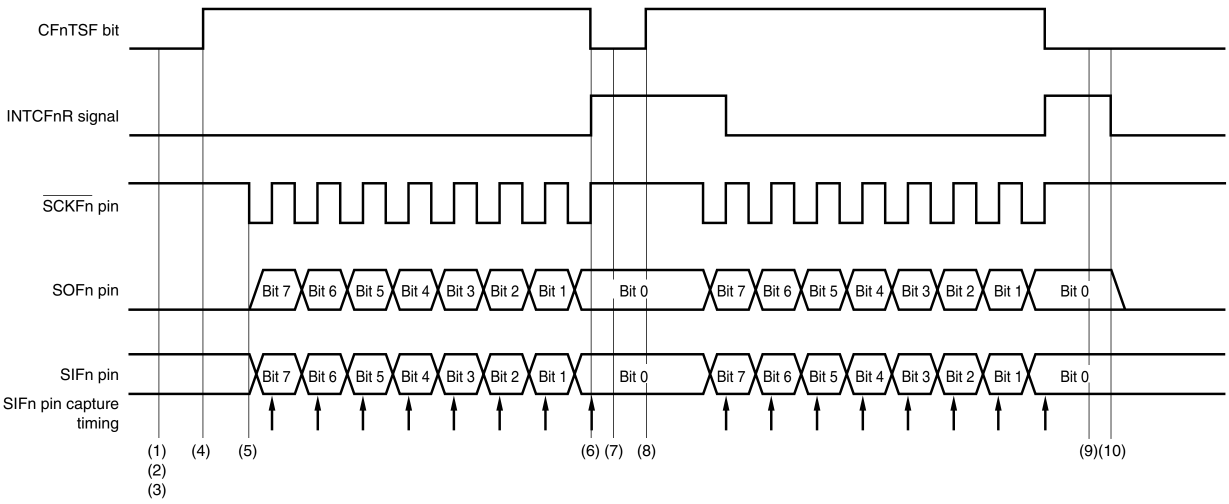


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = $f_{XX}/4$, and master mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CFnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by writing the transmit data to the CFnTX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{\text{SCKFn}}$ pin, output the transmit data to the SOFn pin in synchronization with the serial clock, and capture the receive data of the SIFn pin.

(6) When transmission/reception of the transfer data length set with the CFnCTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception end interrupt request signal (INTCFnR) at the last edge of the serial clock, and clear the CFnTSF bit to 0.

(7) Read the CFnRX register.

(8) To continue transmission/reception, write the transmit data to the CFnTX register again.

(9) Read the CFnRX register.

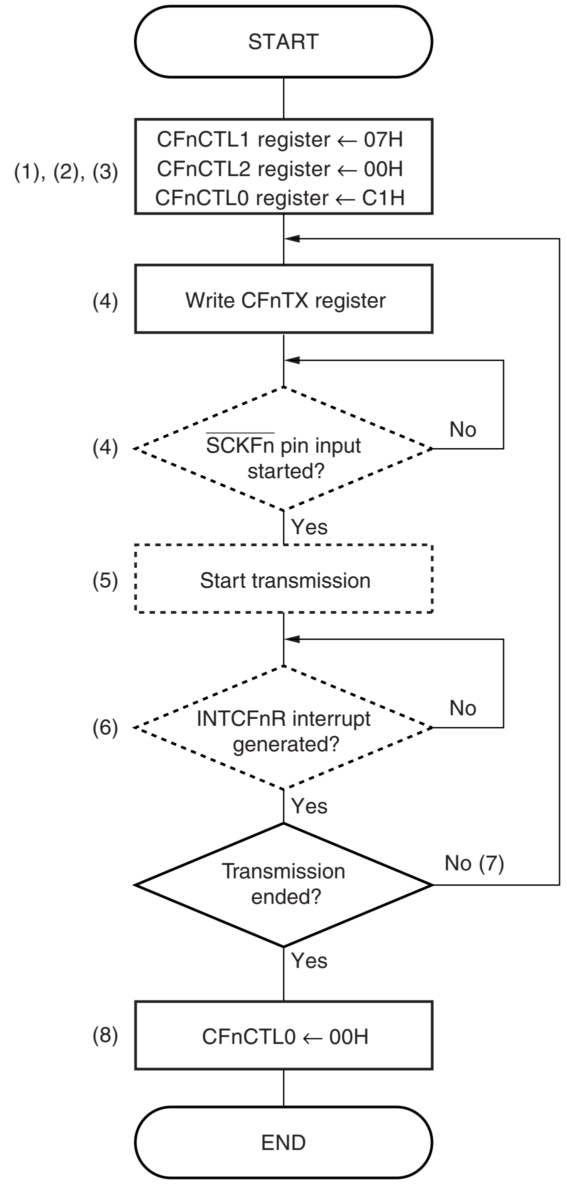
(10) To end transmission/reception, write the CFnCTL0.CFnPWR bit = 0, the CFnCTL0.CFnTXE bit = 0, and the CFnCTL0.CFnRXE bit = 0.

Remark n = 0 to 2

16.5.4 Single transfer mode (slave mode, transmission mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKFn}$) (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 111), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

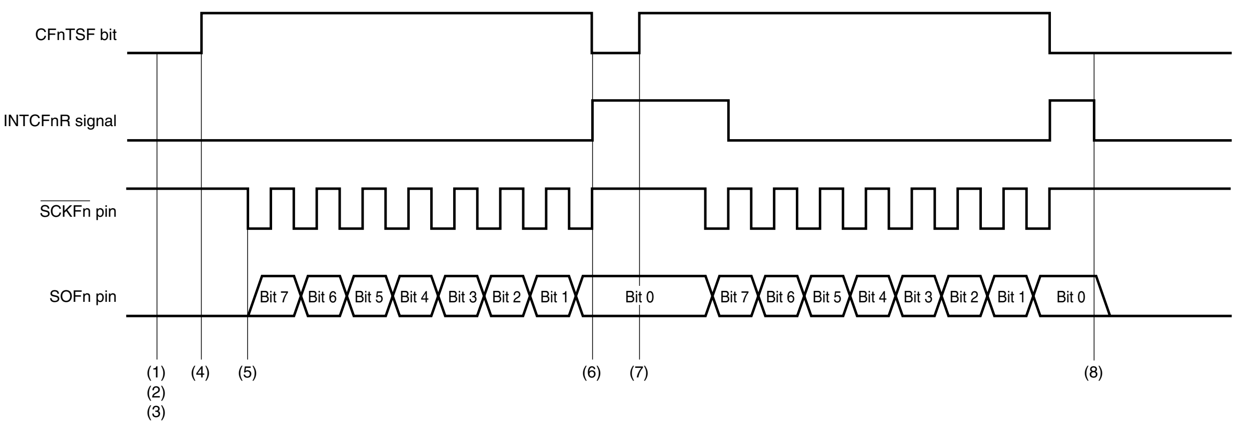


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKFn}$), and slave mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CFnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by writing the transmit data to the CFnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOFn pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CFnCTL2 register is completed, stop the serial clock input and transmit data output, generate the reception end interrupt request signal (INTCFnR) at the last edge of the serial clock, and clear the CFnTSF bit to 0.

(7) To continue transmission, write the transmit data to the CFnTX register again after the INTCFnR signal is generated, and wait for a serial clock input.

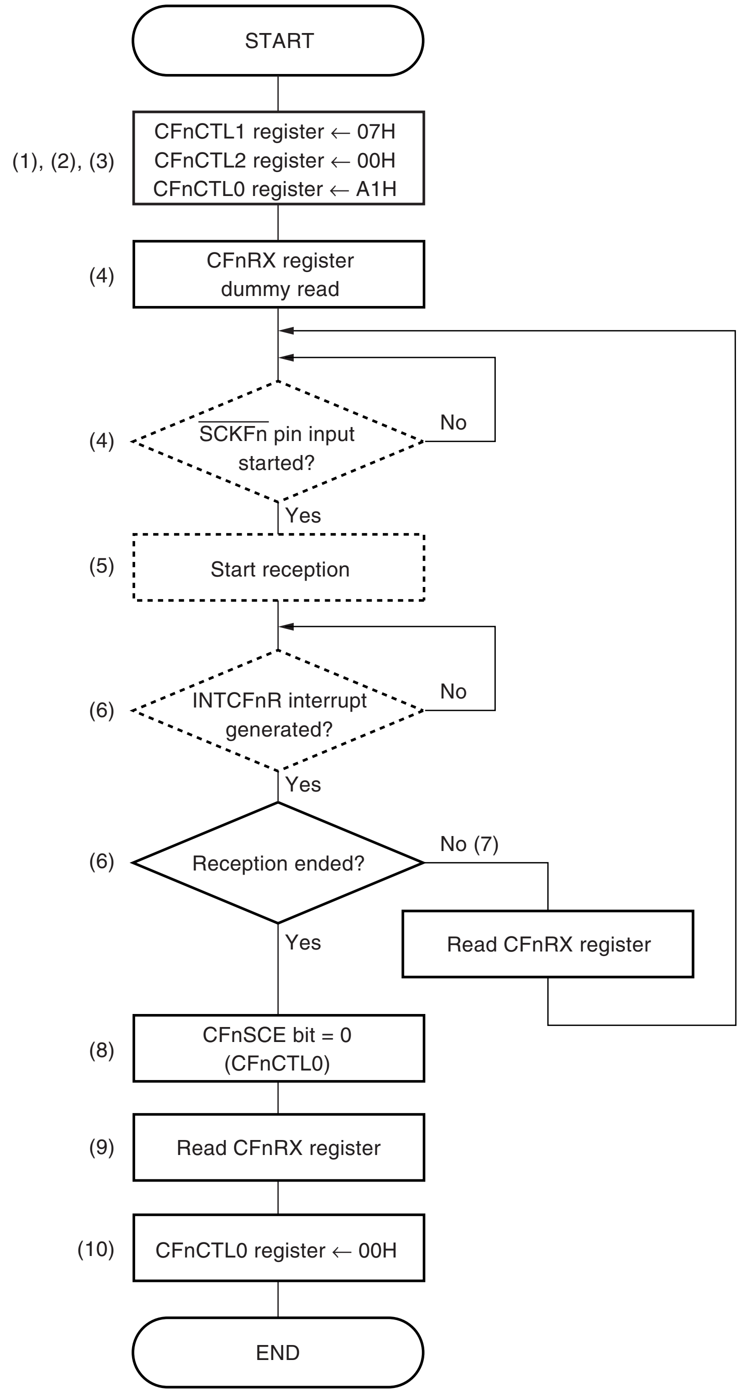
(8) To end transmission, write the CFnCTL0.CFnPWR bit = 0 and the CFnCTL0.CFnTXE bit = 0.

Remark n = 0 to 2

16.5.5 Single transfer mode (slave mode, reception mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{\text{SCKFn}}$) (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 111), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

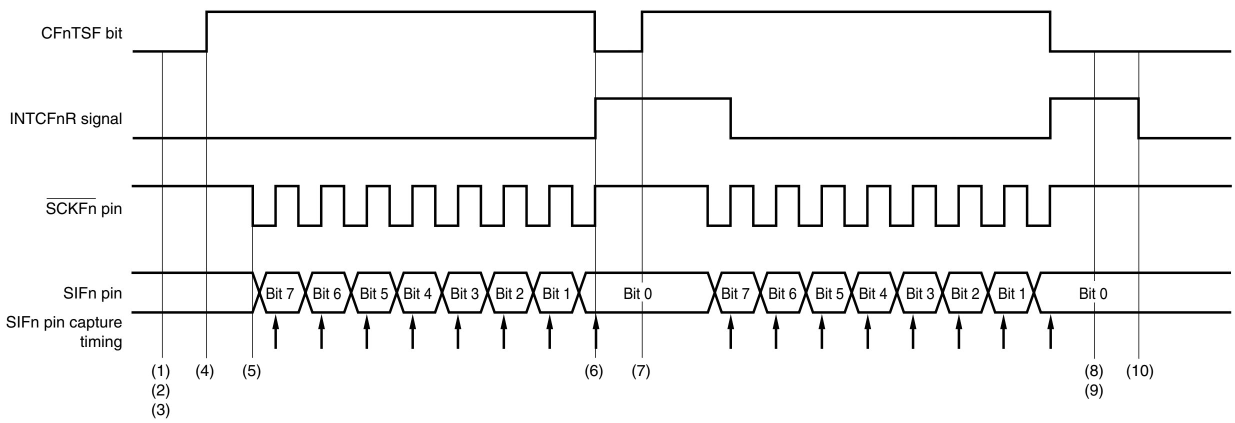


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{\mathrm{SCKFn}}$), and slave mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CFnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by performing a dummy read of the CFnRX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIFn pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CFnCTL2 register is completed, stop the serial clock input and data capturing, generate the reception end interrupt request signal (INTCFnR) at the last edge of the serial clock, and clear the CFnTSF bit to 0.

(7) To continue reception, read the CFnRX register with the CFnCTL0.CFnSCE bit = 1 remained after the INTCFnR signal is generated, and wait for a serial clock input.

(8) To end reception, write the CFnSCE bit = 0.

(9) Read the CFnRX register.

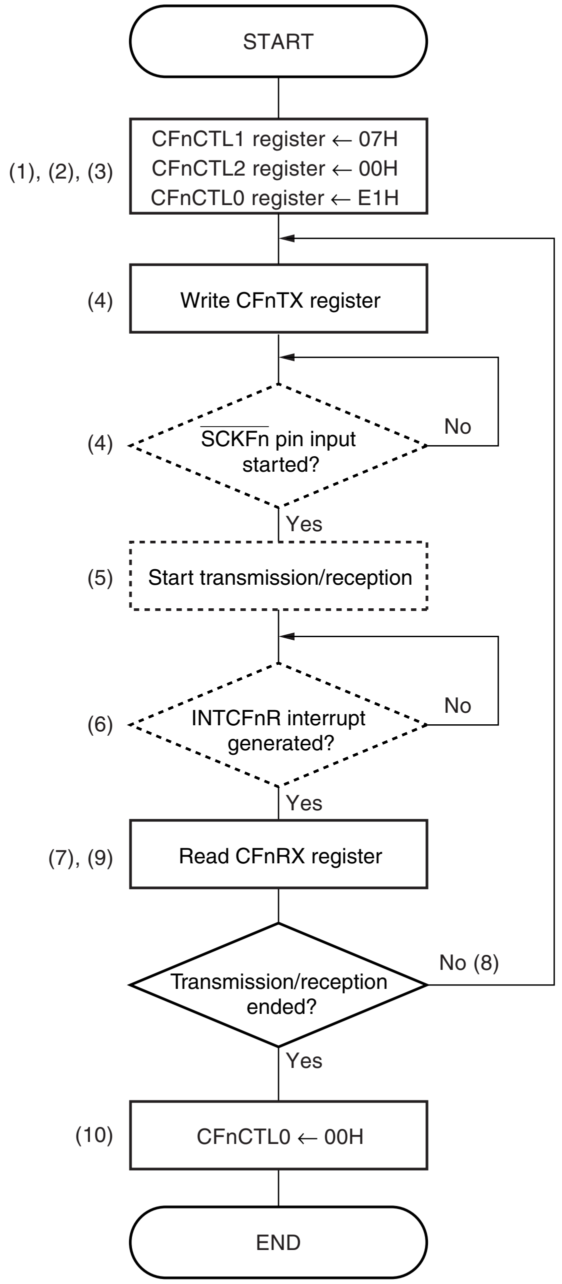
(10) To end reception, write the CFnCTL0.CFnPWR bit = 0 and the CFnCTL0.CFnRXE bit = 0.

Remark n = 0 to 2

16.5.6 Single transfer mode (slave mode, transmission/reception mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{\text{SCKFn}}$) (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 111), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

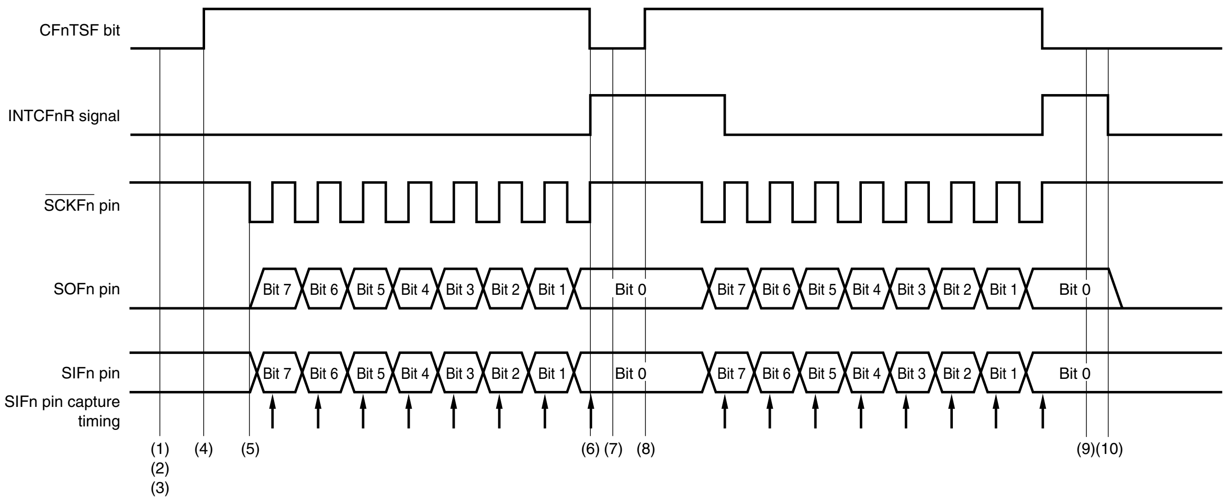


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKFn}$), and slave mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CFnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by writing the transmit data to the CFnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOFn pin in synchronization with the serial clock, and capture the receive data of the SIFn pin.

(6) When transmission/reception of the transfer data length set with the CFnCTL2 register is completed, stop the serial clock input, transmit data output, and data capturing, generate the reception end interrupt request signal (INTCFnR) at the last edge of the serial clock, and clear the CFnTSF bit to 0.

(7) Read the CFnRX register.

(8) To continue transmission/reception, write the transmit data to the CFnTX register again, and wait for a serial clock input.

(9) Read the CFnRX register.

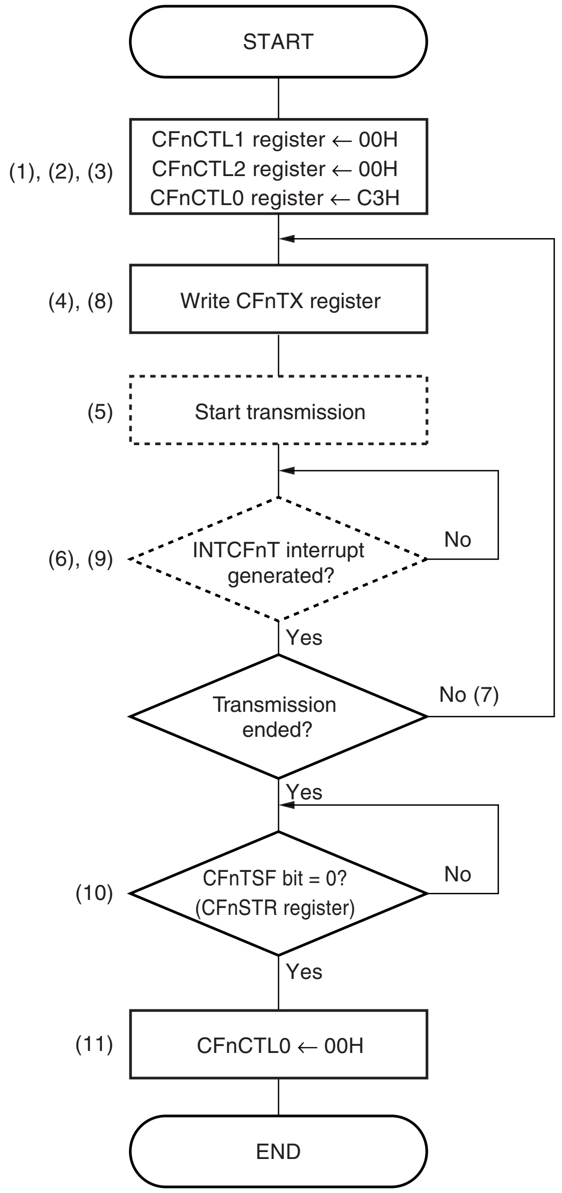
(10) To end transmission/reception, write the CFnCTL0.CFnPWR bit = 0, the CFnCTL0.CFnTXE bit = 0, and the CFnCTL0.CFnRXE bit = 0.

Remark n = 0 to 2

16.5.7 Continuous transfer mode (master mode, transmission mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 000), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

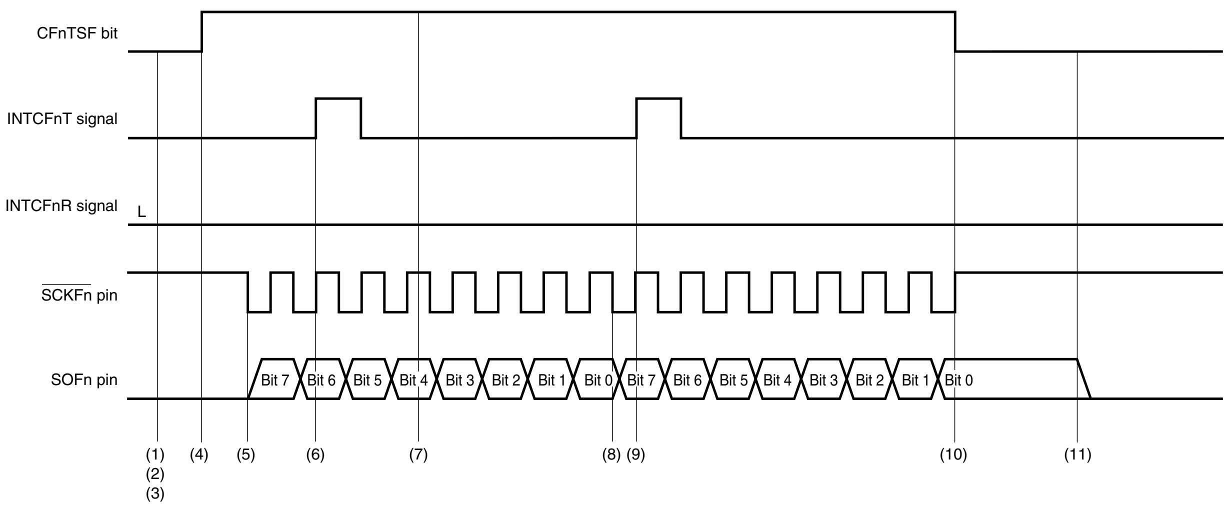


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CFnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by writing the transmit data to the CFnTX register, and transmission is started.

(5) When transmission is started, output the serial clock to the $\overline{\text{SCKFn}}$ pin, and output the transmit data from the SOFn pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CFnTX register to the shift register is ended and writing to the CFnTX register is enabled, the transmission enable interrupt request signal (INTCFnT) is generated.

(7) To continue transmission, write the transmit data to the CFnTX register again after the INTCFnT signal is generated.

(8) When a new transmit data is written to the CFnTX register before communication end, the next communication is started following communication end.

(9) The transfer of the transmit data from the CFnTX register to the shift register is ended and the INTCFnT signal is generated. To end continuous transmission at the current transmission, do not write to the CFnTX register.

(10) When the next transmit data is not written to the CFnTX register before transfer end, stop the serial clock output to the $\overline{\text{SCKFn}}$ pin after transfer end, and clear the CFnTSF bit to 0.

(11) To release the transmission enable status, write the CFnCTL0.CFnPWR bit = 0 and the CFnCTL0.CFnTXE bit = 0 after checking that the CFnTSF bit = 0.

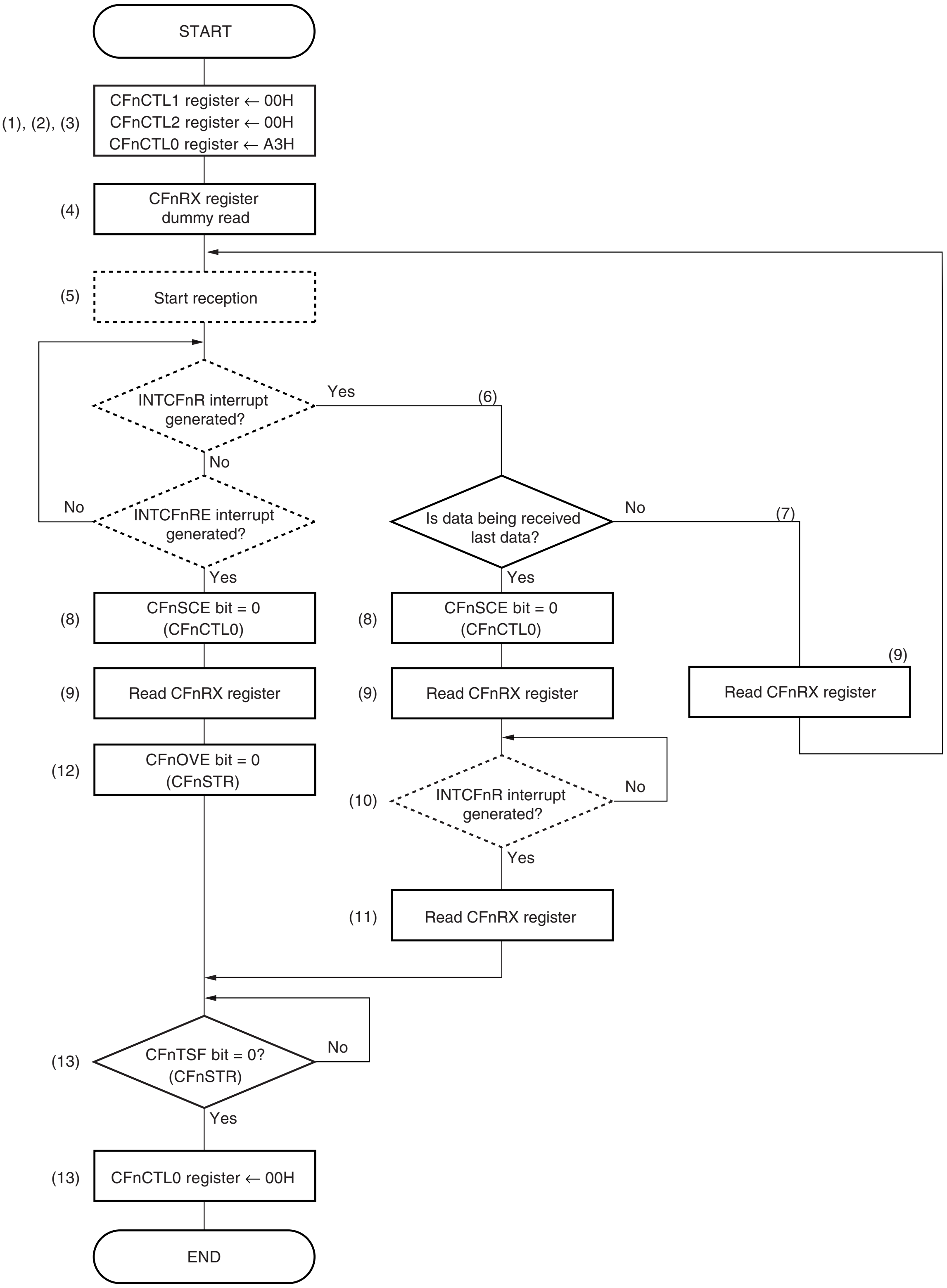
Caution In continuous transmission mode, the reception end interrupt request signal (INTCFnR) is not generated.

Remark n = 0 to 2

16.5.8 Continuous transfer mode (master mode, reception mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 000), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

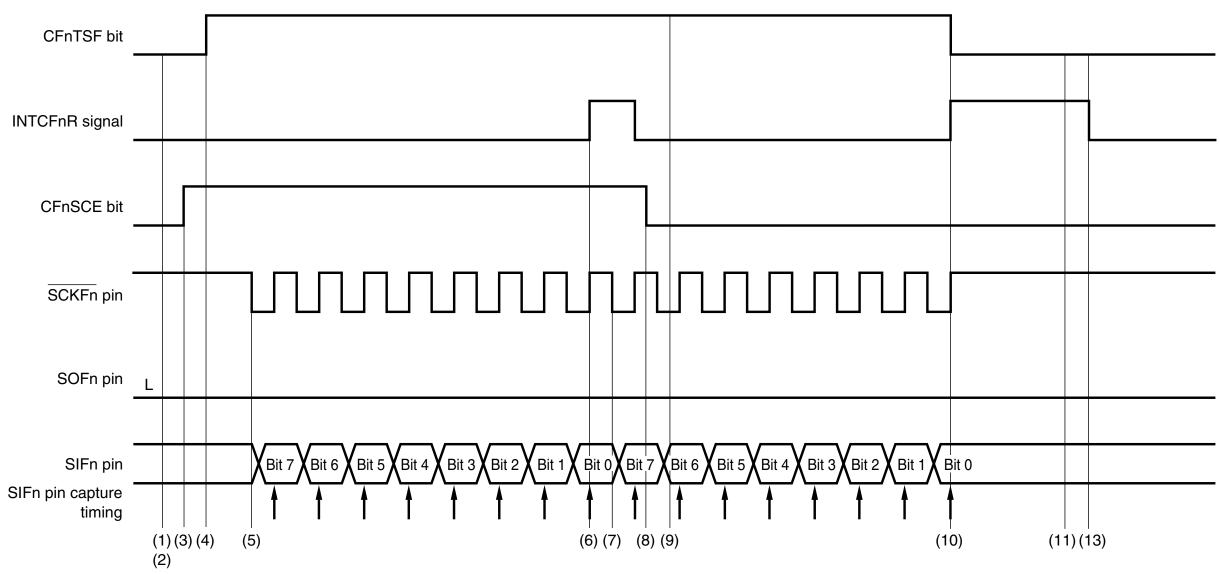


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A3H to the CFnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by performing a dummy read of the CFnRX register, and reception is started.

(5) When reception is started, output the serial clock to the $\overline{\text{SCKFn}}$ pin, and capture the receive data of the SIFn pin in synchronization with the serial clock.

(6) When reception is ended, the reception end interrupt request signal (INTCFnR) is generated, and reading of the CFnRX register is enabled.

(7) When the CFnCTL0.CFnSCE bit = 1 upon communication end, the next communication is started following communication end.

(8) To end continuous reception at the current reception, write the CFnSCE bit = 0.

(9) Read the CFnRX register.

(10) When reception is ended, the INTCFnR signal is generated, and reading of the CFnRX register is enabled. When the CFnSCE bit = 0 is set before communication end, stop the serial clock output to the $\overline{\text{SCKFn}}$ pin, and clear the CFnTSF bit to 0, to end the receive operation.

(11) Read the CFnRX register.

(12) If an overrun error occurs, write the CFnSTR.CFnOVE bit = 0, and clear the error flag.

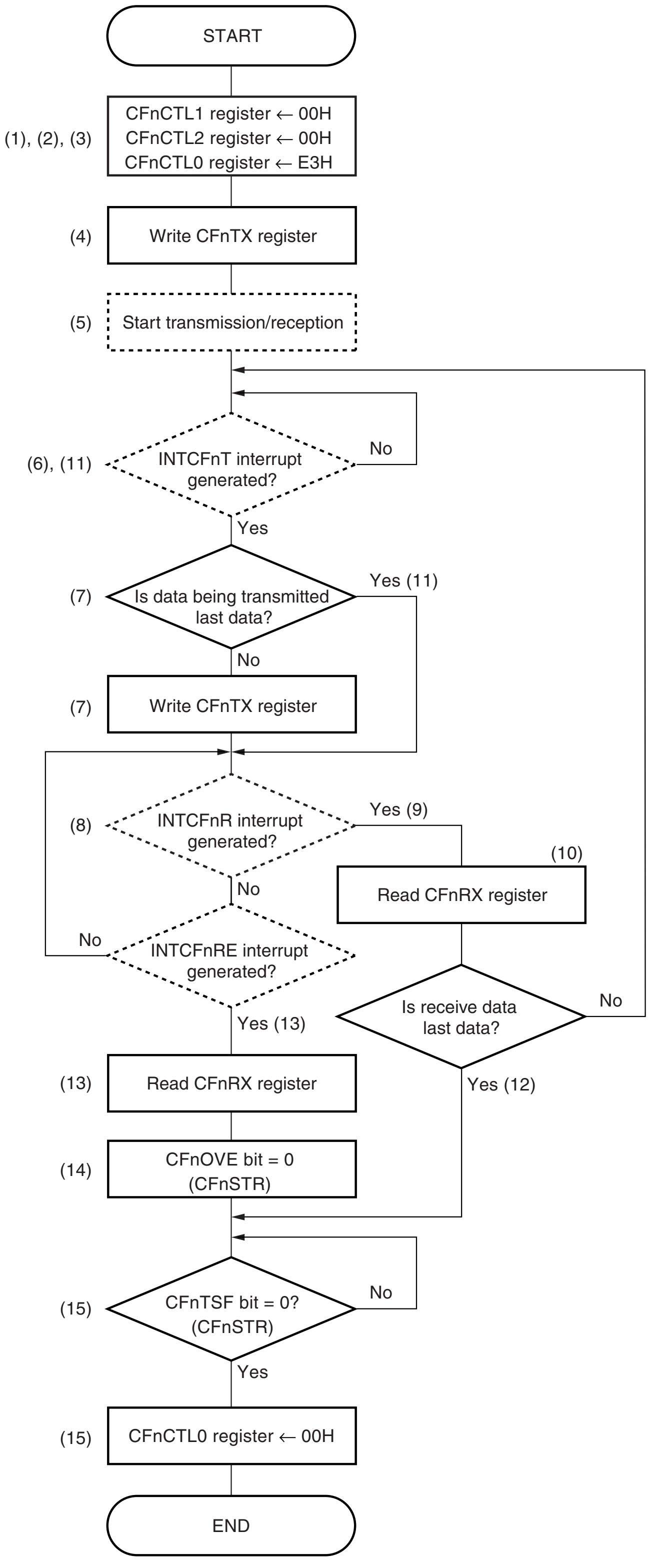
(13) To release the reception enable status, write the CFnCTL0.CFnPWR bit = 0 and the CFnCTL0.CFnRXE bit = 0 after checking that the CFnTSF bit = 0.

Remark n = 0 to 2

16.5.9 Continuous transfer mode (master mode, transmission/reception mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 000), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow



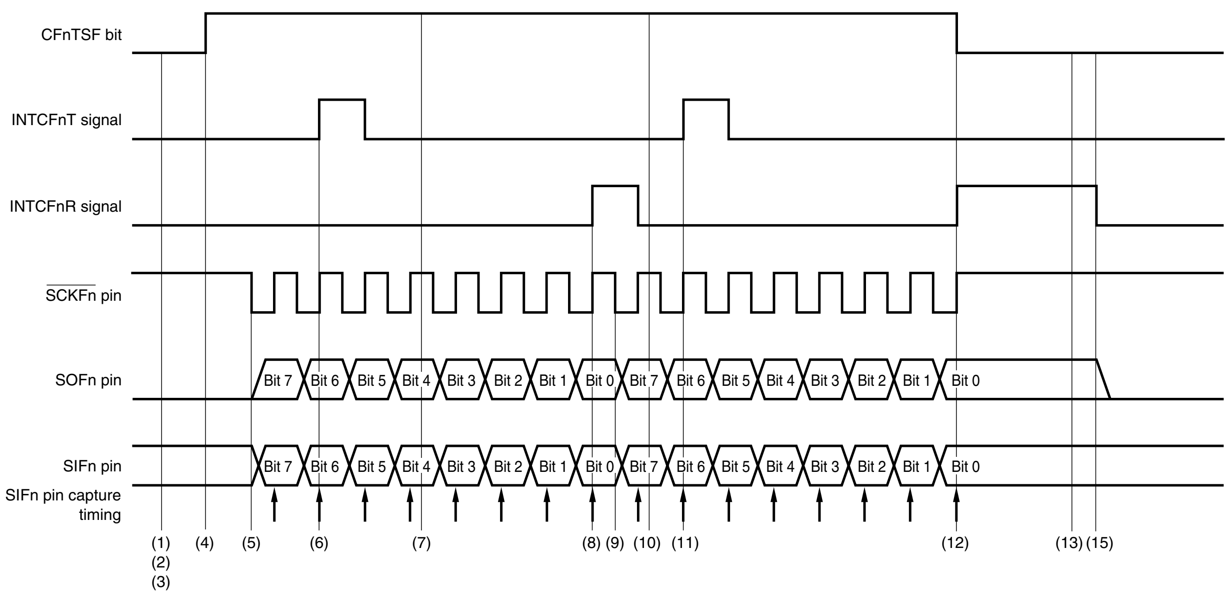
Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing

(1/2)



(1) Write 00H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CFnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by writing the transmit data to the CFnTX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{\text{SCKFn}}$ pin, output the transmit data to the SOFn pin in synchronization with the serial clock, and capture the receive data of the SIFn pin.

(6) When transfer of the transmit data from the CFnTX register to the shift register is ended and writing to the CFnTX register is enabled, the transmission enable interrupt request signal (INTCFnT) is generated.

(7) To continue transmission/reception, write the transmit data to the CFnTX register again after the INTCFnT signal is generated.

(8) When one transmission/reception is ended, the reception end interrupt request signal (INTCFnR) is generated, and reading of the CFnRX register is enabled.

(9) When a new transmit data is written to the CFnTX register before communication end, the next communication is started following communication end.

(10) Read the CFnRX register.

Remark n = 0 to 2

(2/2)

(11) The transfer of the transmit data from the CFnTX register to the shift register is ended and the INTCFnT signal is generated. To end continuous transmission/reception at the current transmission/reception, do not write to the CFnTX register.

(12) When the next transmit data is not written to the CFnTX register before transfer end, stop the serial clock output to the $\overline{\text{SCKFn}}$ pin after transfer end, and clear the CFnTSF bit to 0.

(13) When the reception error interrupt request signal (INTCFnRE) is generated, read the CFnRX register.

(14) If an overrun error occurs, write the CFnSTR.CFnOVE bit = 0, and clear the error flag.

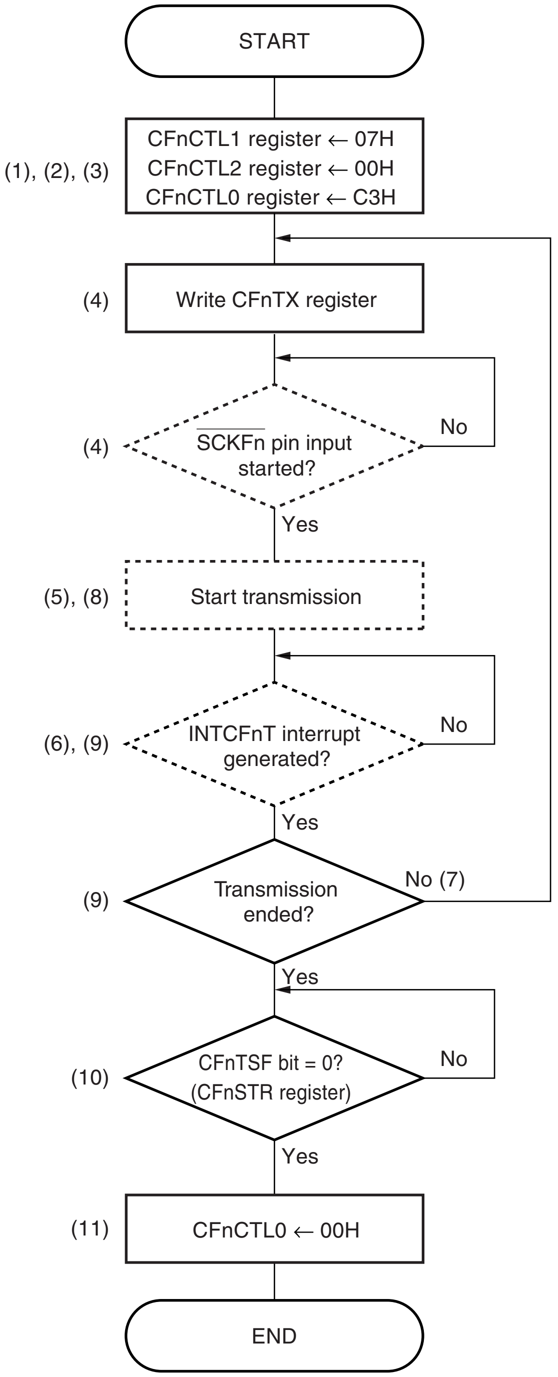
(15) To release the transmission/reception enable status, write the CFnCTL0.CFnPWR bit = 0, the CFnCTL0.CFnTXE bit = 0, and the CFnCTL0.CFnRXE bit = 0 after checking that the CFnTSF bit = 0.

Remark n = 0 to 2

16.5.10 Continuous transfer mode (slave mode, transmission mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKFn}$) (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 111), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

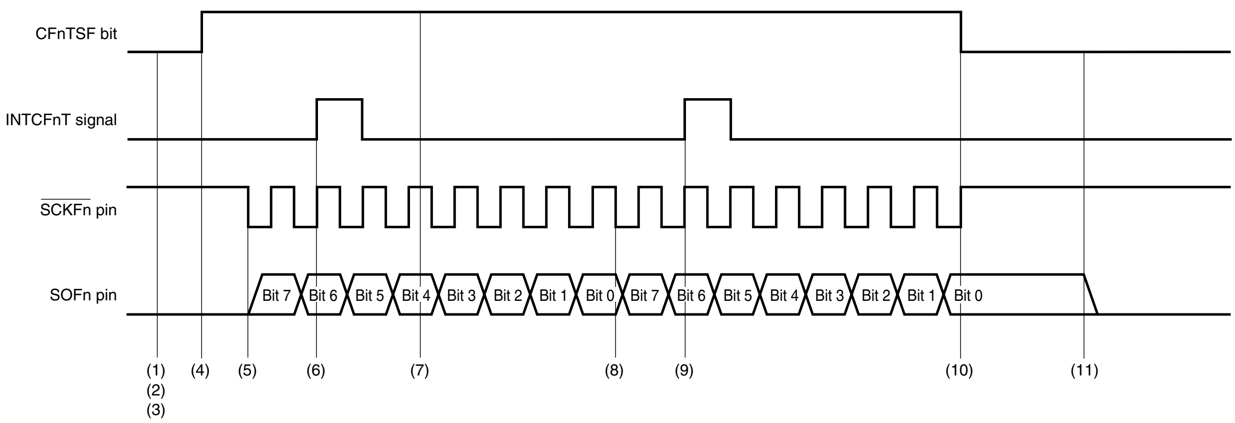


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKFn}$), and slave mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CFnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by writing the transmit data to the CFnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOFn pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CFnTX register to the shift register is ended and writing to the CFnTX register is enabled, the transmission enable interrupt request signal (INTCFnT) is generated.

(7) To continue transmission, write the transmit data to the CFnTX register again after the INTCFnT signal is generated.

(8) When a serial clock is input following end of the transmission of the transfer data length set with the CFnCTL2 register, continuous transmission is started.

(9) When transfer of the transmit data from the CFnTX register to the shift register is ended and writing to the CFnTX register is enabled, the INTCFnT signal is generated. To end continuous transmission at the current transmission, do not write to the CFnTX register.

(10) When the clock of the transfer data length set with the CFnCTL2 register is input without writing to the CFnTX register, clear the CFnTSF bit to 0 to end transmission.

(11) To release the transmission enable status, write the CFnCTL0.CFnPWR bit = 0 and the CFnCTL0.CFnTXE bit = 0 after checking that the CFnTSF bit = 0.

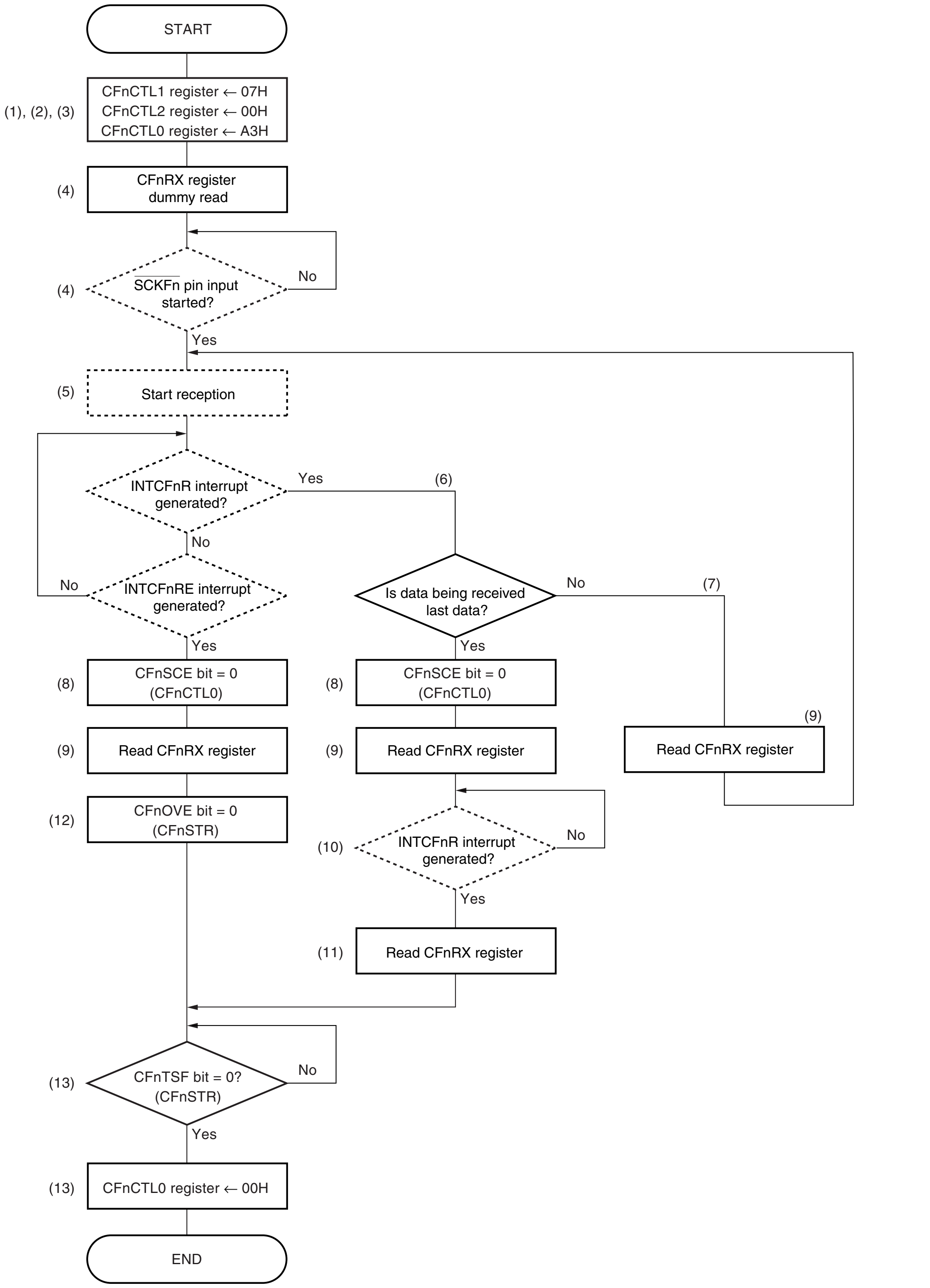
Caution In continuous transmission mode, the reception end interrupt request signal (INTCFnR) is not generated.

Remark n = 0 to 2

16.5.11 Continuous transfer mode (slave mode, reception mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{\text{SCKFn}}$) (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 111), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow

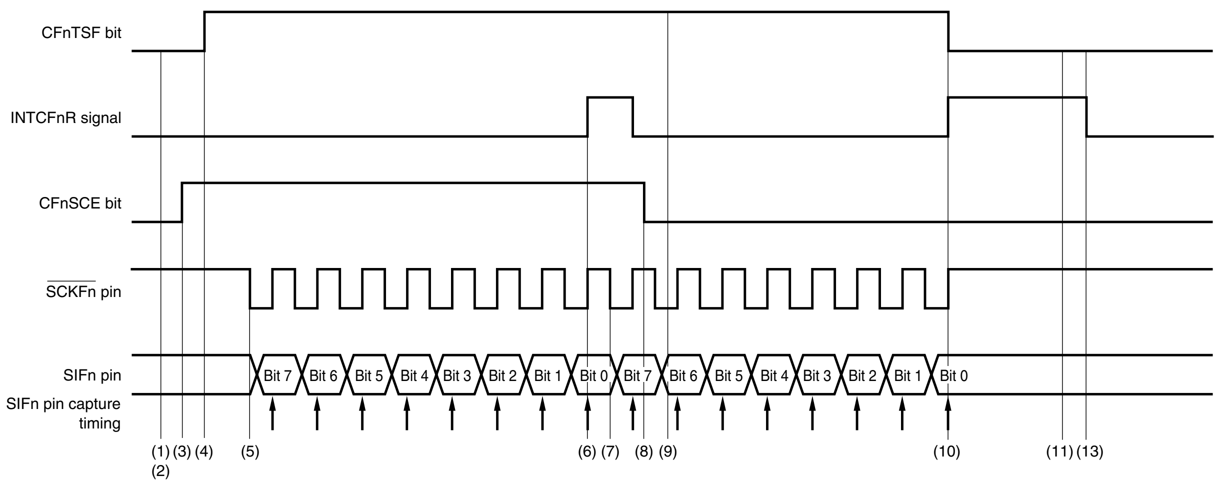


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{\text{SCKFn}}$), and slave mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A3H to the CFnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by performing a dummy read of the CFnRX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIFn pin in synchronization with the serial clock.

(6) When reception is ended, the reception end interrupt request signal (INTCFnR) is generated, and reading of the CFnRX register is enabled.

(7) When a serial clock is input in the CFnCTL0.CFnSCE bit = 1 status, continuous reception is started.

(8) To end continuous reception at the current reception, write the CFnSCE bit = 0.

(9) Read the CFnRX register.

(10) When reception is ended, the INTCFnR signal is generated, and reading of the CFnRX register is enabled. When the CFnSCE bit = 0 is set before communication end, clear the CFnTSF bit to 0 to end the receive operation.

(11) Read the CFnRX register.

(12) If an overrun error occurs, write the CFnSTR.CFnOVE bit = 0, and clear the error flag.

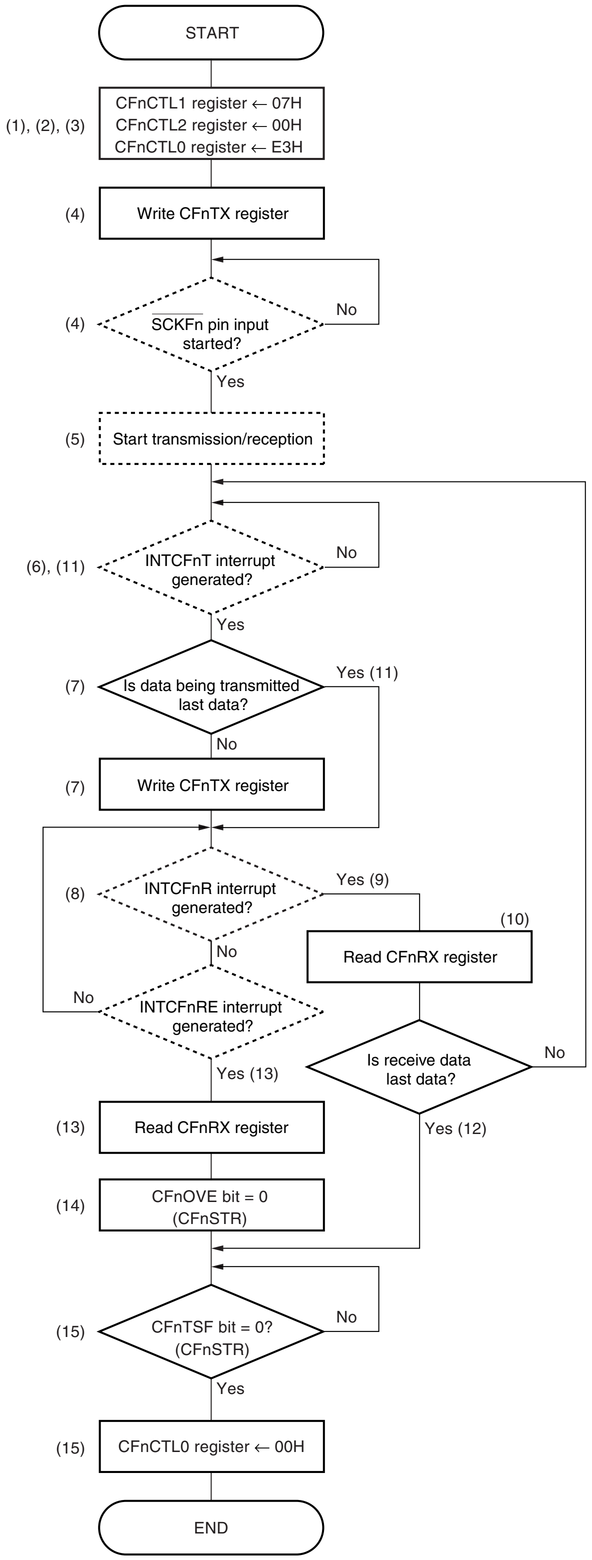
(13) To release the reception enable status, write the CFnCTL0.CFnPWR bit = 0 and the CFnCTL0.CFnRXE bit = 0 after checking that the CFnTSF bit = 0.

Remark n = 0 to 2

16.5.12 Continuous transfer mode (slave mode, transmission/reception mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{\text{SCKFn}}$) (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 111), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow



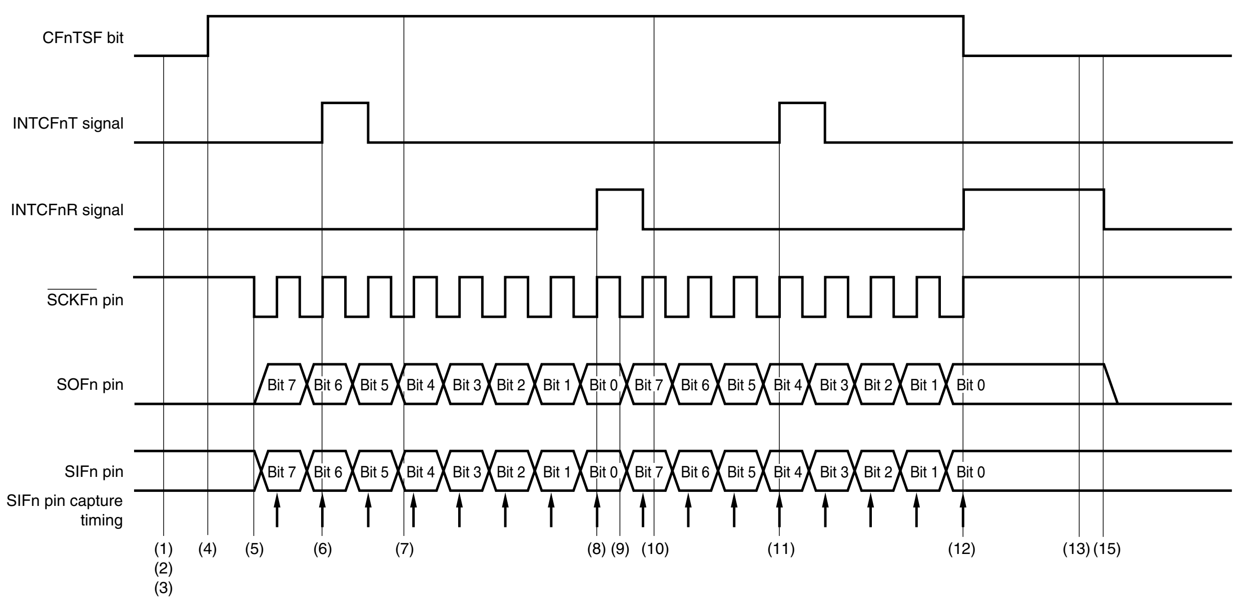
Remarks **1.** The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing

(1/2)



(1) Write 07H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKFn}$), and slave mode.

(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CFnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CFnSTR.CFnTSF bit is set to 1 by writing the transmit data to the CFnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOFn pin in synchronization with the serial clock, and capture the receive data of the SIFn pin.

(6) When transfer of the transmit data from the CFnTX register to the shift register is ended and writing to the CFnTX register is enabled, the transmission enable interrupt request signal (INTCFnT) is generated.

(7) To continue transmission, write the transmit data to the CFnTX register again after the INTCFnT signal is generated.

(8) When reception of the transfer data length set with the CFnCTL2 register is completed, the reception end interrupt request signal (INTCFnR) is generated, and reading of the CFnRX register is enabled.

(9) When a serial clock is input continuously, continuous transmission/reception is started.

(10) Read the CFnRX register.

(11) When transfer of the transmit data from the CFnTX register to the shift register is ended and writing to the CFnTX register is enabled, the INTCFnT signal is generated. To end continuous transmission/reception at the current transmission/reception, do not write to the CFnTX register.

Remark n = 0 to 2

(2/2)

(12) When the clock of the transfer data length set with the CFnCTL2 register is input without writing to the CFnTX register, the INTCFnR signal is generated. Clear the CFnTSF bit to 0 to end transmission/reception.

(13) When the reception error interrupt request signal (INTCFnRE) is generated, read the CFnRX register.

(14) If an overrun error occurs, write the CFnSTR.CFnOVE bit = 0, and clear the error flag.

(15) To release the transmission/reception enable status, write the CFnCTL0.CFnPWR bit = 0, the CFnCTL0.CFnTXE bit = 0, and the CFnCTL0.CFnRXE bit = 0 after checking that the CFnTSF bit = 0.

Remark n = 0 to 2

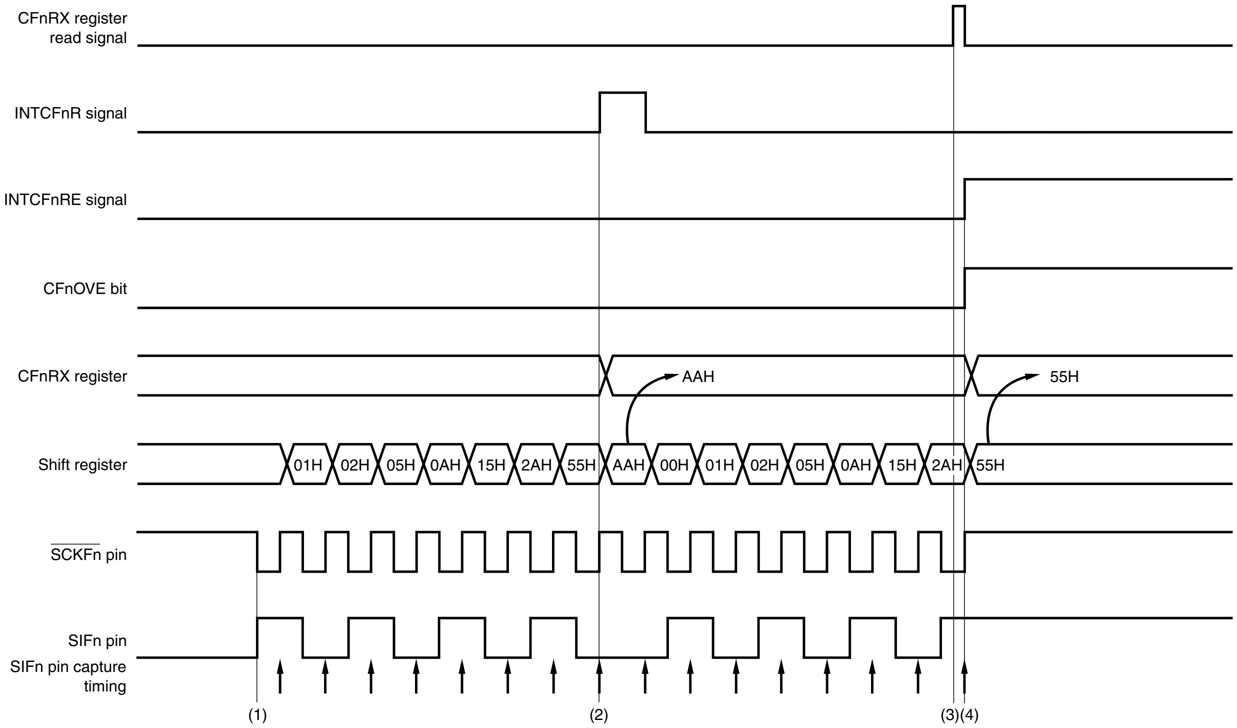
16.5.13 Reception error

When transfer is performed with reception enabled (CFnCTL0.CFnRXE bit = 1) in the continuous transfer mode, the reception error interrupt request signal (INTCFnRE) is generated when the next receive operation is ended before the CFnRX register is read after the reception end interrupt request signal (INTCFnR) is generated, and the overrun error flag (CFnSTR.CFnOVE) is set to 1.

Even if an overrun error has occurred, the previous receive data is lost since the CFnRX register is updated. Even if a reception error has occurred, the INTCFnRE signal is generated again upon the next reception end if the CFnRX register is not read.

To avoid an overrun error, end reading the CFnRX register until one half clock before sampling the last bit of the next receive data from the INTCFnR signal generation.

(1) Operation timing



(1) Start continuous transfer.

(2) End of the first transfer

(3) The CFnRX register cannot be read until one half-clock before the end of the second transfer.

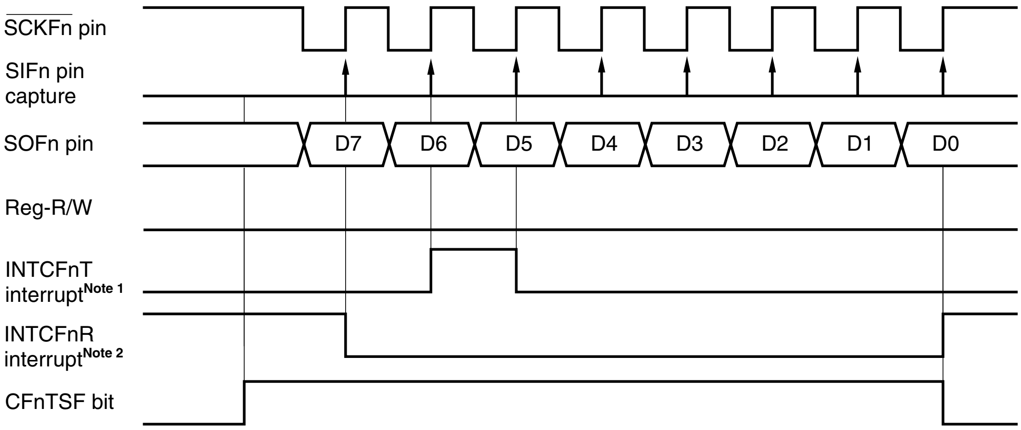
(4) When an overrun error occurs and the reception error interrupt request signal (INTCFnRE) is generated, the overrun error flag (CFnSTR.CFnOVE) is set (1). The receive data is overwritten.

Remark n = 0 to 2

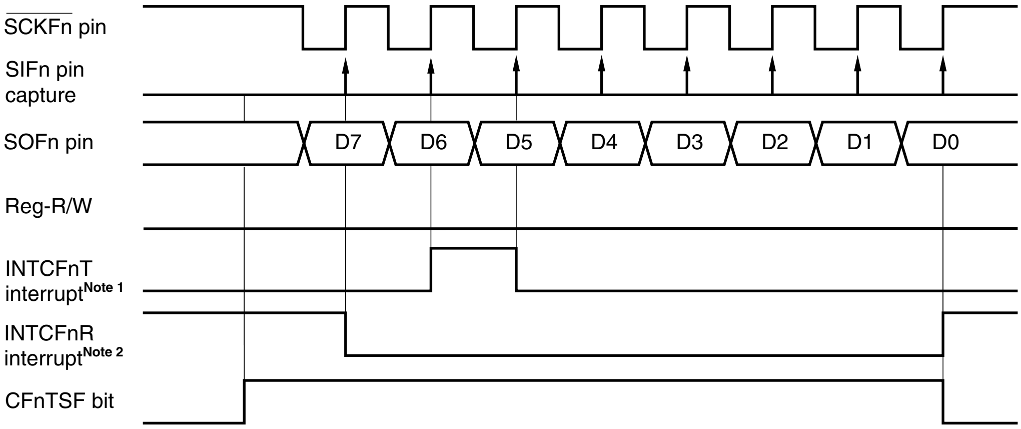
16.5.14 Clock timing

(1/2)

(i) Communication type 1 (CFnCKP and CFnDAP bits = 00)



(ii) Communication type 3 (CFnCKP and CFnDAP bits = 10)



Notes 1. The INTCFnT interrupt is set when the data written to the CFnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCFnT interrupt request signal is not generated, but the INTCFnR interrupt request signal is generated upon end of communication.

2. The INTCFnR interrupt occurs if reception is correctly ended and receive data is ready in the CFnRX register while reception is enabled. In the single mode, the INTCFnR interrupt request signal is generated even in the transmission mode, upon end of communication.

Caution **In single transfer mode, writing to the CFnTX register with the CFnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.**

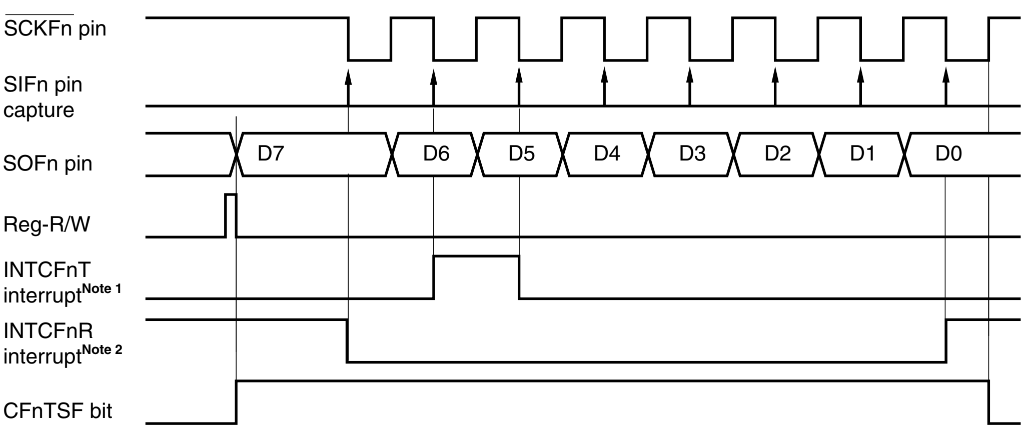
For example, if the next data is written to the CFnTX register when DMA is started by generating the INTCFnR signal, the written data is not transferred because the CFnTSF bit is set to 1.

Use the continuous transfer mode, not the single transfer mode, for such applications.

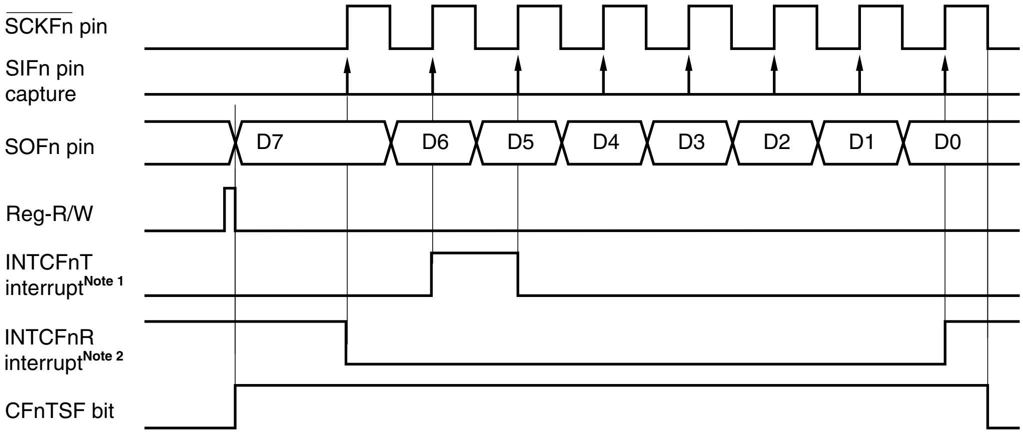
Remark n = 0 to 2

(2/2)

(iii) Communication type 2 (CFnCKP and CFnDAP bits = 01)



(iv) Communication type 4 (CFnCKP and CFnDAP bits = 11)



Notes 1. The INTCFnT interrupt is set when the data written to the CFnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCFnT interrupt request signal is not generated, but the INTCFnR interrupt request signal is generated upon end of communication.

2. The INTCFnR interrupt occurs if reception is correctly ended and receive data is ready in the CFnRX register while reception is enabled. In the single mode, the INTCFnR interrupt request signal is generated even in the transmission mode, upon end of communication.

Caution **In single transfer mode, writing to the CFnTX register with the CFnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.**

For example, if the next data is written to the CFnTX register when DMA is started by generating the INTCFnR signal, the written data is not transferred because the CFnTSF bit is set to 1.

Use the continuous transfer mode, not the single transfer mode, for such applications.

Remark n = 0 to 2

16.6 Output Pins

(1) $\overline{\text{SCKFn}}$ pin

When CSIFn operation is disabled (CFnCTL0.CFnPWR bit = 0), the $\overline{\text{SCKFn}}$ pin output status is as follows.

Remark n = 0 to 2

CFnCKP	CFnCKS2	CFnCKS1	CFnCKS0	$\overline{\text{SCKFn}}$ Pin Output
0	1	1	1	High impedance
	Other than above			Fixed to high level
1	1	1	1	High impedance
	Other than above			Fixed to low level

Remark The output level of the $\overline{\text{SCKFn}}$ pin changes if any of the CFnCTL1.CFnCKP and CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits is rewritten.

(2) SOFn pin

When CSIFn operation is disabled (CFnPWR bit = 0), the SOFn pin output status is as follows.

Remark n = 0 to 2

CFnTXE	CFnDAP	CFnDIR	SOFn Pin Output
0	×	×	Fixed to low level
1	0	×	SOFn latch value (low level)
	1	0	CFnTX value (MSB)
		1	CFnTX value (LSB)

Remarks 1. The SOFn pin output changes when any one of the CFnCTL0.CFnTXE, CFnCTL0.CFnDIR, or CFnCTL1.CFnDAP bits is rewritten.

2. ×: 0 or 1

CHAPTER 17 I^{2}C BUS

To use the I^{2}C bus function, use the P30/SCL and P31/SDA pins as the serial transmit/receive data and set them to N-ch open-drain output.

In the V850E/IG4-H and V850E/IH4-H, one channel of I^{2}C bus is provided.

17.1 Features

The I^{2}C has the following two modes.

- Operation stop mode
- I^{2}C (Inter IC) bus mode (multimaster supported)

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^{2}C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL) line and a serial data bus (SDA) line.

This mode complies with the I^{2}C bus format and the master device can generate “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received state and data by hardware. This function can simplify the part of application program that controls the I^{2}C bus.

Since the SCL and SDA pins are used for N-ch open drain outputs, I^{2}C requires pull-up resistors for the serial clock line and the serial data bus line.

17.2 Configuration

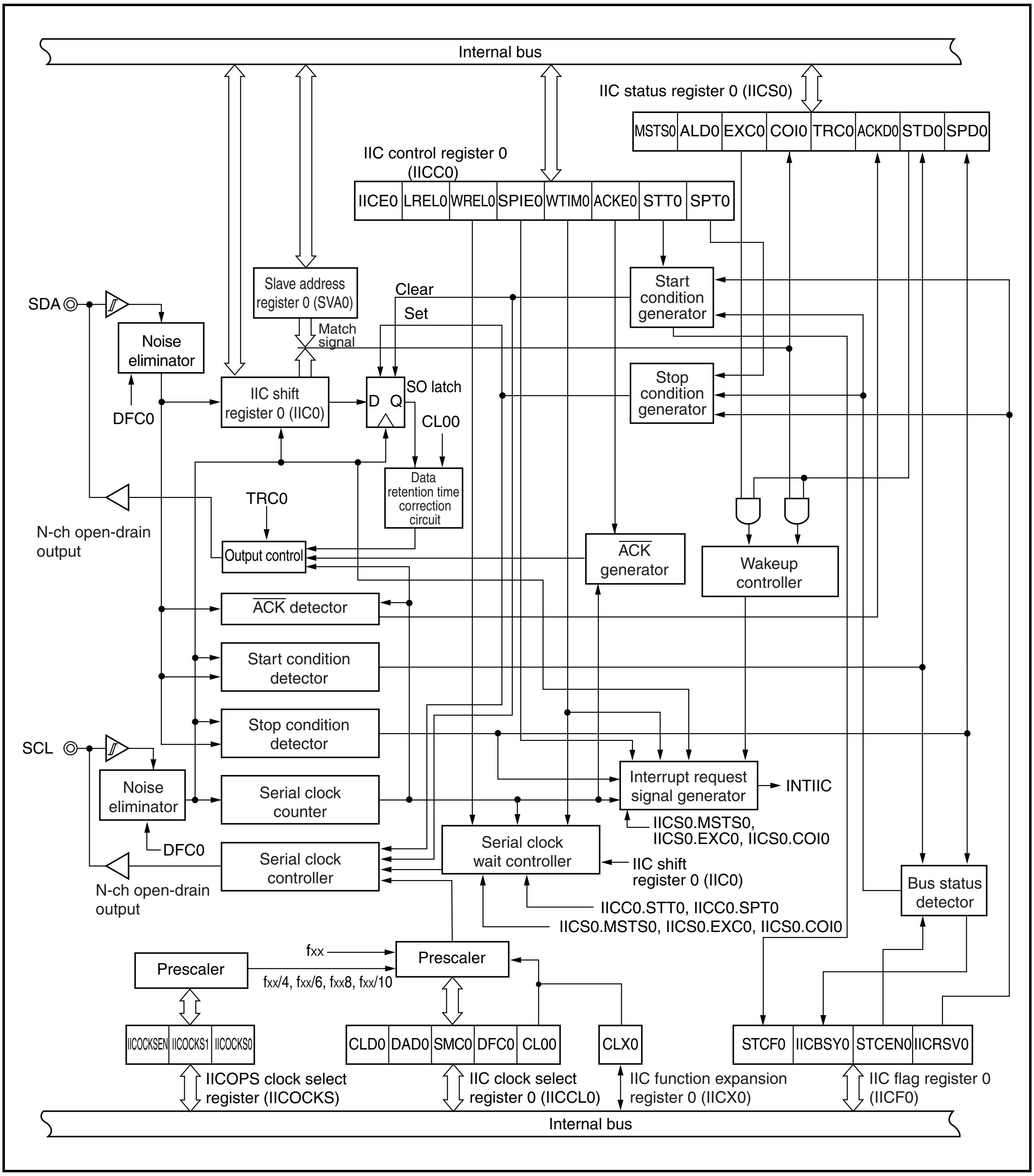
I²C includes the following hardware.

Table 17-1. Configuration of I²C

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC flag register 0 (IICF0) IIC clock select register 0 (IICCL0) IIC function expansion register 0 (IICX0) IICOPS clock select register (IICOCKS)

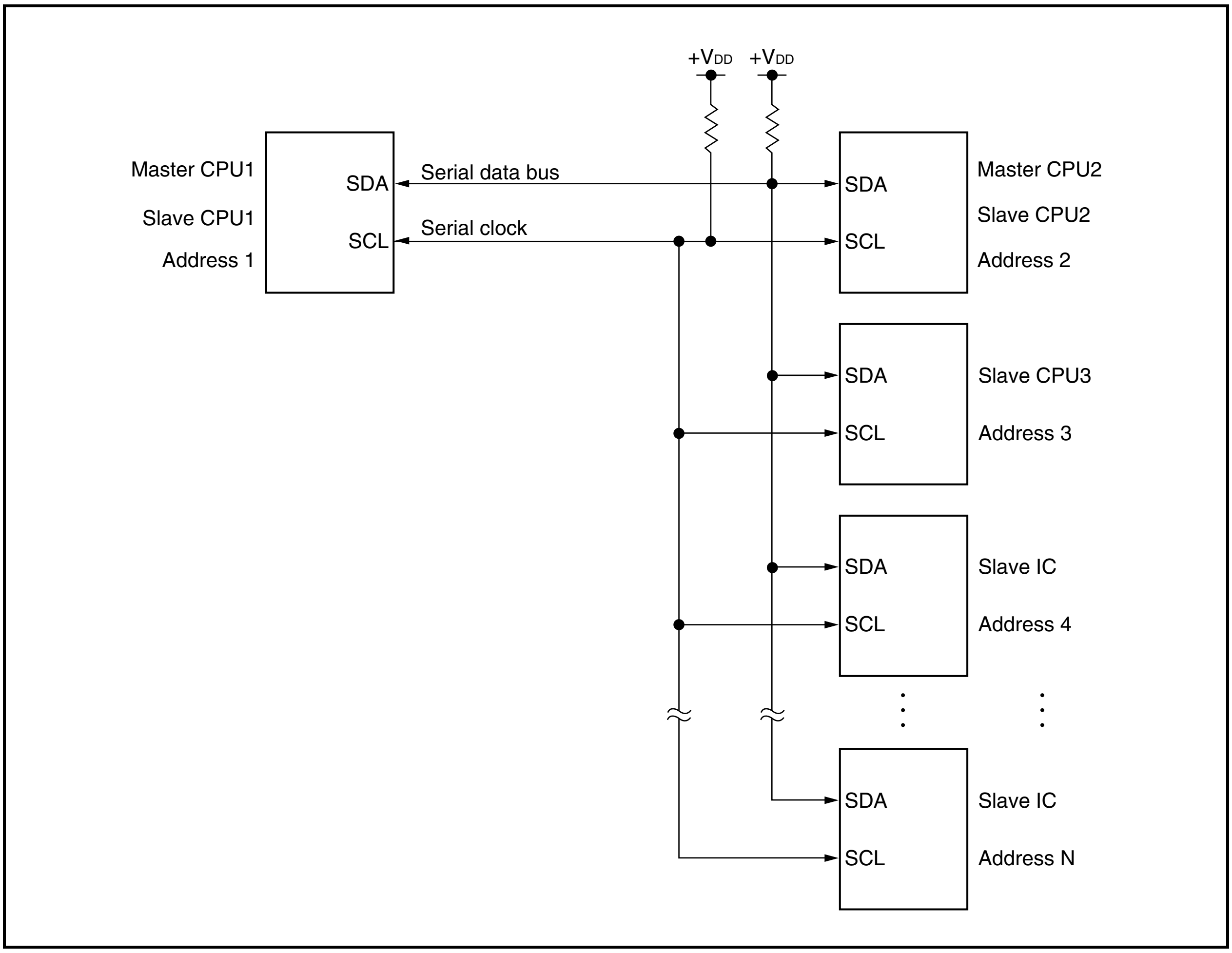
A block diagram of I^{2}C is shown below.

Figure 17-1. Block Diagram of I^{2}C



A serial bus configuration example is shown below.

Figure 17-2. Serial Bus Configuration Example Using I^{2}C Bus



(1) IIC shift register 0 (IIC0)

The IIC0 register is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data. The IIC0 register can be used for both transmission and reception.

Write and read operations to the IIC0 register are used to control the actual transmit and receive operations.

The IIC0 register can be read or written in 8-bit units.

Reset sets IIC0 to 00H.

(2) Slave address register 0 (SVA0)

The SVA0 register sets local addresses when in slave mode.

The SVA0 register can be read or written in 8-bit units.

Reset sets SVA0 to 00H.

(3) SO latch

The SO latch is used to retain the SDA pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request signal (INTIIC) when the address received by this register matches the address value set to the SVA0 register or when an extension code is received.

(5) Prescaler

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC).
An I^{2}C interrupt is generated following either of two triggers.

- Falling of the eighth or ninth clock of the serial clock (set by IICC0.WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by IICC0.SPIE0 bit)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) $\overline{\text{ACK}}$ generator, stop condition detector, start condition detector, and $\overline{\text{ACK}}$ detector

These circuits are used to generate and detect various statuses.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(12) Start condition generator

This circuit generates a start condition when the IICC0.STT0 bit is set.
However, in the communication reservation disabled status (IICF0.IICRSV0 bit = 1), when the bus is not released (IICF0.IICBSY0 bit = 1), start condition requests are ignored and the IICF0.STCF0 bit is set to 1.

(13) Stop condition generator

A stop condition is generated when the IIC0.SPT0 bit is set (1).

(14) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.
However, as the bus status cannot be detected immediately following operation, the initial status is set by the IICF0.STCEN0 bit.

17.2.1 Pin functions of each channel

The SCL and SDA pins used by I^{2}C in the V850E/IG4-H and V850E/IH4-H are alternately used for other functions as shown in Table 17-2. To use these pins for I^{2}C, set up the related registers as described in **Table 4-16 Settings When Pins Are Used for Alternate Functions**.

Table 17-2. Pins Used by I^{2}C

Pin No.		Port	I^{2}C Serial Clock I/O	I^{2}C Serial Transmission/ Reception Data I/O	Other Functions
IG4-H	IH4-H				
GC	GF				
54	106	P30	SCL	–	RXDA1/$\overline{\text{WR1}}$
55	107	P31	–	SDA	TXDA1/$\overline{\text{WAIT}}$

Remark IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

17.3 Mode Switching Between I^{2}C and UARTA1

In the V850E/IG4-H and V850E/IH4-H, I^{2}C and UARTA1 share a pin, and these functions cannot be used at the same time. To use the pin for the I^{2}C function, set up the PMC3 and PFC3 registers in advance.

Switching the operation mode between I^{2}C and UARTA1, the serial interfaces, is described below.

Caution The operations related to transmission and reception of I^{2}C or UARTA1 are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 17-3. Mode Switch Settings of I^{2}C and UARTA1

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	0	0	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC3n	PFC3n	Operation mode
0	×	Port I/O mode
1	0	UART1 mode
1	1	I^{2}C mode

Remarks 1. n = 0, 1

2. × = 0 or 1

17.4 Registers

I^{2}C is controlled by the following registers.

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC flag register 0 (IICF0)
- IIC clock select register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)
- IICOPS clock select register (IICOCKS)

The following registers are also used.

- IIC shift register 0 (IIC0)
- Slave address register 0 (SVA0)

Remark For the alternate-function pin settings, see **Table 4-16 Settings When Pins Are Used for Alternate Functions**.

(1) IIC control register 0 (IICC0)

The IICC0 register is used to enable/stop I^{2}C operations, set wait timing, and set other I^{2}C operations.

The IICC0 register can be read or written in 8-bit or 1-bit units. However, set the SPIE0, WTIM0, and ACKE0 bits when the IICE0 bit is 0 or during the wait period. When setting the IICE0 bit from “0” to “1”, these bits can also be set at the same time.

Reset sets this register to 00H.

(1/4)

After reset: 00H R/W Address: FFFFFD82H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^{2}C operation enable/disable specification
0	Stop operation. Reset the IICS0 register[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit to 1 when the SCL and SDA lines are high level.	
Condition for clearing (IICE0 bit = 0)	Condition for setting (IICE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0[Note 2]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL and SDA lines are set to high impedance. The STT0, SPT0, IICS0.MSTS0, IICS0.EXC0, IICS0.COI0, IICS0.TRC0, IICS0.ACKD0, and IICS0.STD0 bits are cleared to 0.
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL0 bit = 0)	Condition for setting (LREL0 bit = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICS0 register, and the IICF0.STCF0, IICF0.IICBSY0, IICCL0.CLD0, and IICCL0.DAD0 bits are reset.

2. This flag’s signal is invalid when the IICE0 bit = 0.

Caution If the I^{2}C operation is enabled (IICE0 bit = 1) when the SCL line is high level and the SDA line is low level, the start condition is detected immediately. To avoid this, after enabling the I^{2}C operation, immediately set the LREL0 bit to 1 with a bit manipulation instruction.

(2/4)

WREL0[Note]	Wait cancellation control
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared to 0 after wait is canceled.
Condition for clearing (WREL0 bit = 0)	Condition for setting (WREL0 bit = 1)
• Automatically cleared after execution • Reset	• Set by instruction

SPIE0[Note]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
Condition for clearing (SPIE0 bit = 0)	Condition for setting (SPIE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[Note]	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.
An interrupt is generated at the falling of the 9th clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after $\overline{\text{ACK}}$ is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIM0 bit = 0)	Condition for setting (WTIM0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0[Note]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDA line is set to low level.
The ACKE0 bit setting is invalid for address reception. In this case, $\overline{\text{ACK}}$ is generated when the addresses match. However, the ACKE0 bit setting is valid for address reception of the extension code.	
Condition for clearing (ACKE0 bit = 0)	Condition for setting (ACKE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

Note This flag's signal is invalid when the IICE0 bit = 0.

(3/4)

STT0	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in STOP mode): Generate a start condition (for starting as master). The SDA line is changed from high level to low level while the SCL line is high level and then the start condition is generated. Next, after the rated amount of time has elapsed, the SCL line is changed to low level (wait status). When a third party is communicating • When communication reservation function is enabled (IICF0.IICRSV0 bit = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV0 bit = 1) The IICF0.STCF0 bit is set to 1 and the information set (1) to the STT0 bit is cleared. No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing

For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.

For master transmission: A start condition may not be generated normally during the $\overline{\text{ACK}}$ period. Set to 1 during the wait period that follows output of the ninth clock.

- Cannot be set to 1 at the same time as the SPT0 bit.
- When the STT0 bit is set to 1, setting the STT0 bit to 1 again is disabled until the setting is cleared to 0.

Condition for clearing (STT0 bit = 0)	Condition for setting (STT0 bit = 1)
• When the STT0 bit is set to 1 in the communication reservation disabled status • Cleared by loss in arbitration • Cleared when start condition is generated by master device • When the LREL0 bit = 1 (exit from communications) • When the IICE0 bit = 0 (operation stop) • Reset	• Set by instruction

Remark The STT0 bit is 0 if it is read after data setting.

(4/4)

<table>
<tr><th>SPT0</th><th>Stop condition trigger</th></tr>
<tr><td>0</td><td>Stop condition is not generated.</td></tr>
<tr><td>1</td><td>Stop condition is generated (termination of master device’s transfer).
After the SDA line goes to low level, either set the SCL line to high level or wait until the SCL pin goes to high level. Next, after the rated amount of time has elapsed, the SDA line is changed from low level to high level and a stop condition is generated.</td></tr>
<tr><td colspan="2">Cautions concerning setting timing
For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and during the wait period after slave has been notified of final reception.
For master transmission: A stop condition may not be generated normally during the $\overline{\text{ACK}}$ period. Set to 1 during the wait period that follows output of the ninth clock.
• Cannot be set to 1 at the same time as the STT0 bit.
• The SPT0 bit can be set to 1 only when in master mode[Note].
• When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock.
The WTIM0 bit should be changed from 0 to 1 during the wait period following output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows output of the ninth clock.
• When the SPT0 bit is set to 1, setting the SPT0 bit to 1 again is disabled until the setting is cleared to 0.</td></tr>
<tr><td>Condition for clearing (SPT0 bit = 0)</td><td>Condition for setting (SPT0 bit = 1)</td></tr>
<tr><td>• Cleared by loss in arbitration
• Automatically cleared after stop condition is detected
• When the LREL0 bit = 1 (exit from communications)
• When the IICE0 bit = 0 (operation stop)
• Reset</td><td>• Set by instruction</td></tr>
</table>

Note Set the SPT0 bit to 1 only in master mode. However, the SPT0 bit must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to operation enable status. For details, see **17.15 Cautions**.

Caution When the IICS0.TRC0 bit is set to 1, the WREL0 bit is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared to 0 and the SDA line is set to high impedance.

Remark The SPT0 bit is 0 if it is read after data setting.

(2) IIC status register 0 (IICS0)

The IICS0 register indicates the status of the I^{2}C bus.

The IICS0 register is read-only, in 8-bit or 1-bit units.

However, the IICS0 register can only be read when the IICC0.STT0 bit is 1 or during the wait period.

Reset sets this register to 00H.

(1/3)

After reset: 00H R Address: FFFFFD86H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

<table>
<tr><td>MSTS0</td><td colspan="2">Master device status</td></tr>
<tr><td>0</td><td colspan="2">Slave device status or communication standby status</td></tr>
<tr><td>1</td><td colspan="2">Master device communication status</td></tr>
<tr><td colspan="2">Condition for clearing (MSTS0 bit = 0)</td><td>Condition for setting (MSTS0 bit = 1)</td></tr>
<tr><td colspan="2">• When a stop condition is detected
• When the ALD0 bit = 1 (arbitration loss)
• Cleared by the IICC0.LREL0 bit = 1 (exit from communications)
• When the IICC0.IICE0 bit changes from 1 to 0 (operation stop)
• Reset</td><td>• When a start condition is generated</td></tr>
</table>

<table>
<tr><td>ALD0</td><td colspan="2">Detection of arbitration loss</td></tr>
<tr><td>0</td><td colspan="2">This status means either that there was no arbitration or that the arbitration result was a “win”.</td></tr>
<tr><td>1</td><td colspan="2">This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared to 0.</td></tr>
<tr><td colspan="2">Condition for clearing (ALD0 bit = 0)</td><td>Condition for setting (ALD0 bit = 1)</td></tr>
<tr><td colspan="2">• Automatically cleared after the IICS0 register is read[Note]
• When the IICE0 bit changes from 1 to 0 (operation stop)
• Reset</td><td>• When the arbitration result is a “loss”.</td></tr>
</table>

Note This bit is also cleared when a bit manipulation instruction is executed for another bit in the IICS0 register.

(2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.
Condition for clearing (EXC0 bit = 0)	Condition for setting (EXC0 bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.
Condition for clearing (COI0 bit = 0)	Condition for setting (COI0 bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 • Reset	• When the received address matches the local address (SVA0 register) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA line is set for high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDA line (valid starting at the rising edge of the first byte’s ninth clock).
Condition for clearing (TRC0 bit = 0)	Condition for setting (TRC0 bit = 1)
• When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by the IICC0.WREL0 bit = 1[Note] (wait release) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset Master • When “1” is output to the first byte’s LSB (transfer direction specification bit) Slave • When a start condition is detected When not used for communication	Master • When a start condition is generated • When “0” is output to the first byte’s LSB (transfer direction specification bit) Slave • When “1” is input in the first byte’s LSB (transfer direction specification bit)

Note The IICS0.TRC0 bit is cleared to 0 and the SDA line become high impedance when the IICC0.WREL0 bit is set to 1 and wait state is released at the ninth clock with the TRC0 bit = 1.

(3/3)

ACKD0	Detection of $\overline{\text{ACK}}$
0	$\overline{\text{ACK}}$ was not detected.
1	$\overline{\text{ACK}}$ was detected.
Condition for clearing (ACKD0 bit = 0)	Condition for setting (ACKD0 bit = 1)
• When a stop condition is detected • At the rising edge of the next byte’s first clock • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• After the SDA pin is set to low level at the rising edge of the SCL pin’s ninth clock

STD0	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect
Condition for clearing (STD0 bit = 0)	Condition for setting (STD0 bit = 1)
• When a stop condition is detected • At the rising edge of the next byte’s first clock following address transfer • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device’s communication is terminated and the bus is released.
Condition for clearing (SPD0 bit = 0)	Condition for setting (SPD0 bit = 1)
• At the rising edge of the address transfer byte’s first clock following setting of this bit and detection of a start condition • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

(3) IIC flag register 0 (IICF0)

IICF0 is a register that set the operation mode of I^{2}C and indicate the status of the I^{2}C bus.

These registers can be read or written in 8-bit or 1-bit units. However, the STCF0 and IICBSY0 bits are read-only.

The IICRSV0 bit can be used to enable/disable the communication reservation function (see **17.14 Communication Reservation**).

The STCEN0 bit can be used to set the initial value of the IICBSY0 bit (see **17.15 Cautions**).

The IICRSV0 and STCEN0 bits can be written only when the operation of I^{2}C is disabled (IICC0.IICE0 bit = 0). When operation is enabled, the IICF0 register can be read.

Reset sets this register to 00H.

After reset: 00H R/W[Note] Address: FFFFFD8AH

	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	IICC0.STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT0 flag

Condition for clearing (STCF0 bit = 0)	Condition for setting (STCF0 bit = 1)
• Clearing by setting the STT0 bit = 1 • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 bit = 1).

IICBSY0	I^2C bus status flag
0	Bus release status (initial communication status when STCEN0 bit = 1)
1	Bus communication status (initial communication status when STCEN0 bit = 0)

Condition for clearing (IICBSY0 bit = 0)	Condition for setting (IICBSY0 bit = 1)
• Detection of stop condition • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICE0 bit when the STCEN0 bit = 0

STCEN0	Initial start enable trigger
1	After operation is enabled (IICE0 bit = 1), enable generation of a start condition upon detection of a stop condition.
	After operation is enabled (IICE0 bit = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0 bit = 0)	Condition for setting (STCEN0 bit = 1)
• Detection of start condition • Reset	• Setting by instruction

IICRSV0	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV0 bit = 0)	Condition for setting (IICRSV0 bit = 1)
• Clearing by instruction • Reset	• Setting by instruction

Note Bits 6 and 7 are read-only bits.

Cautions 1. Write to the STCEN0 bit only when the operation is stopped (IICE0 bit = 0).

2. As the bus release status (IICBSY0 bit = 0) is recognized regardless of the actual bus status when the STCEN0 bit = 1, when generating the first start condition (STT0 bit = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to the IICRSV0 bit only when the operation is stopped (IICE0 bit = 0).

(4) IIC clock select register 0 (IICCL0)

The IICCL0 register is used to set the transfer clock for the I^{2}C bus.

The IICCL0 register can be read or written in 8-bit or 1-bit units. However, the CLD0 and DAD0 bits are read-only. The SMC0 and CL00 bits are set in combination with the IICX0.CLX0, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits (see **17.4 (7) I^{2}C transfer clock setting method**).

Set the IICCL0 register when the IICC0.IICE0 bit = 0.

Reset sets this register to 00H.

After reset: 00H R/W[Note] Address: FFFFFD84H

	7	6	<5>	<4>	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	0	CL00

CLD0	Detection of SCL pin level (valid only when IICC0.IICE0 bit = 1)
0	The SCL pin was detected at low level.
1	The SCL pin was detected at high level.

Condition for clearing (CLD0 bit = 0)	Condition for setting (CLD0 bit = 1)
• When the SCL pin is at low level • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• When the SCL pin is at high level

DAD0	Detection of SDA pin level (valid only when IICE0 bit = 1)
0	The SDA pin was detected at low level.
1	The SDA pin was detected at high level.

Condition for clearing (DAD0 bit = 0)	Condition for setting (DAD0 bit = 1)
• When the SDA pin is at low level • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• When the SDA pin is at high level

SMC0	Operation mode switching
0	Operates in standard mode.
1	Operates in high-speed mode.

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of DFC0 bit set/clear. The digital filter is used for noise elimination in high-speed mode.	

CL00	Communication clock selection	
	Normal mode	High-speed mode
0	Fxx/44	Fxx/24
1	Fxx/86	Fxx/24

Note Bits 4 and 5 are read-only bits.

Remark Fxx: Selection clock

(5) IIC function expansion register 0 (IICX0)

This register sets the function expansion of I^{2}C (valid only in high-speed mode).

This register can be read or written in 8-bit or 1-bit units. The CLX0 bit is set in combination with the IICCL0.SMC0, IICCL0.CL00, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits (see **17.4 (7) I^{2}C transfer clock setting method**).

Set the IICX0 register when the IICC0.IICE0 bit = 0.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFD85H

	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

CLX0	Clock select expansion bit
0	Communicate at transfer rate set by the IICCL0.CL00 bit.
1	Communicate at double transfer rate set by the IICCL0.CL00 bit in high-speed mode .

(6) IICOPS clock select register (IICOCKS)

This register controls the division clock of I^{2}C.

This register can be read or written in 8-bit or 1-bit units. The IICOCKS1 and IICOCKS0 bits are set in combination with the IICCL0.SMC0, IICCL0.CL00, and IICX0.CLX0 bits (see **17.4 (7) I^{2}C transfer clock setting method**).

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFD90H

	7	6	5	4	3	2	1	0
IICOCKS	0	0	0	IICOCKSEN	0	0	IICOCKS1	IICOCKS0

IICOCKSEN	Specification of I^{2}C division clock operation
0	I^{2}C division clock operation stop
1	I^{2}C division clock operation enable

IICOCKS1	IICOCKS0	I^{2}C division clock selection
0	0	f_{xx}/16
0	1	f_{xx}/24
1	0	f_{xx}/32
1	1	f_{xx}/40

(7) I^{2}C transfer clock setting method

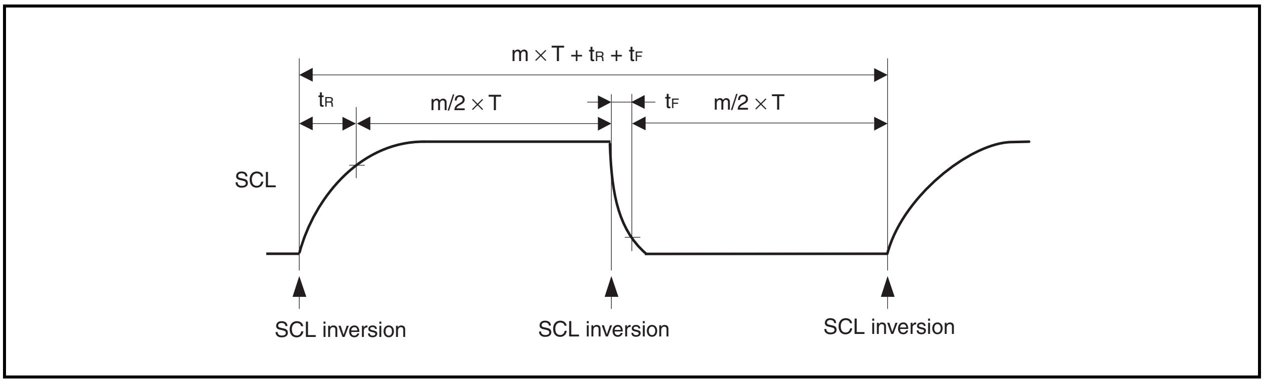
The I^{2}C transfer clock frequency (f_{SCL}) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 288, 384, 576, 768, 1056, 1376, 1408, 1760 (see **Table 17-3 Selection Clock Setting**.)
T: 1/f_{XX}
t_R: SCL rise time
t_F: SCL fall time

For example, the I^{2}C transfer clock frequency (f_{SCL}) when f_{XX} = 100 MHz, m = 576, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(576 \times 10 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 166 \text{ kHz}$$



The selection clock is set using a combination of the IICCL0.SMC0, IICCL0.CL00, IICX0.CLX0, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits.

Table 17-3. Selection Clock Setting

IICX0	IICCL0		Selection Clock	Transfer Clock (f_{XX}/m)	Settable Internal System Clock Frequency (f_{XX}) Range	Operation Mode
Bit 0	Bit 3	Bit 0				
CLX0	SMC0	CL00				
0	0	0	f_{XX}/24 (when IICOCKS = 11H)	f_{XX}/1056	80 MHz to 100 MHz	Normal mode (SMC0 bit = 0)
			f_{XX}/32 (when IICOCKS = 12H)	f_{XX}/1408		
			f_{XX}/40 (when IICOCKS = 13H)	f_{XX}/1760		
0	0	1	f_{XX}/16 (when IICOCKS = 10H)	f_{XX}/1376		
0	1	x	f_{XX}/16 (when IICOCKS = 10H)	f_{XX}/384		High-speed mode (SMC0 bit = 1)
			f_{XX}/24 (when IICOCKS = 11H)	f_{XX}/576	96 MHz to 100 MHz	
			f_{XX}/32 (when IICOCKS = 12H)	f_{XX}/768	100 MHz	
1	0	x	Setting prohibited			
1	1	x	f_{XX}/24 (when IICOCKS = 11H)	f_{XX}/288	96 MHz to 100 MHz	High-speed mode (SMC0 bit = 1)
			f_{XX}/32 (when IICOCKS = 12H)	f_{XX}/384	100 MHz	

Remark 0 or 1

(8) IIC shift register 0 (IIC0)

The IIC0 shift register is used for serial transmission/reception (shift operations) that is synchronized with the serial clock.

The IIC0 shift register can be read or written in 8-bit units, but data should not be written to the IIC0 shift register during a data transfer.

Access (read/write) the IIC0 shift register only during the wait period. Accessing this register in communication states other than the wait period is prohibited. However, for the master device, the IIC0 shift register can be written once only after the transmission trigger bit (IICC0.STT0 bit) has been set to 1.

When the IIC0 shift register is written during wait, the wait is cancelled and data transfer is started.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFD80H

	7	6	5	4	3	2	1	0
IIC0								

(9) Slave address register 0 (SVA0)

The SVA0 register holds the I^2C bus's slave addresses.

However, rewriting this register is prohibited when the IICS0.STD0 bit = 1 (start condition detection).

The SVA0 register can be read or written in 8-bit units, but bit 0 is fixed to 0.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFD83H

	7	6	5	4	3	2	1	0
SVA0								0

17.5 Functions

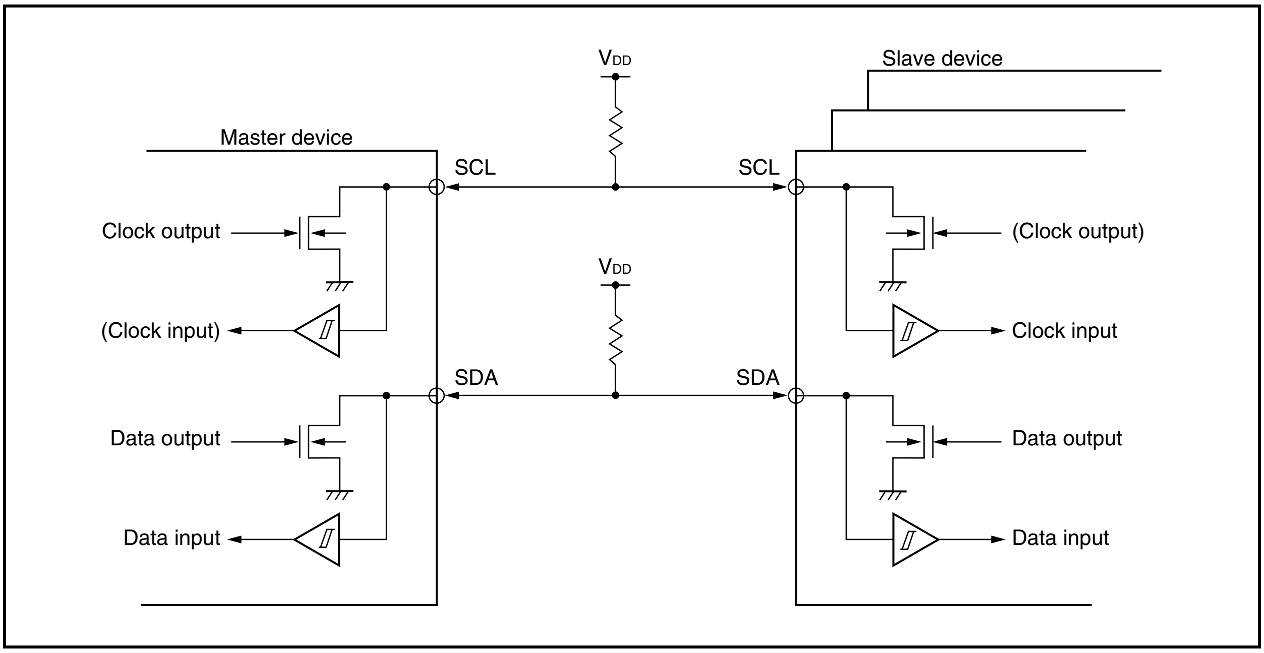
17.5.1 Pin configuration

The serial clock pin (SCL) and serial data bus pin (SDA) are configured as follows.

SCL ....This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
SDA ....This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

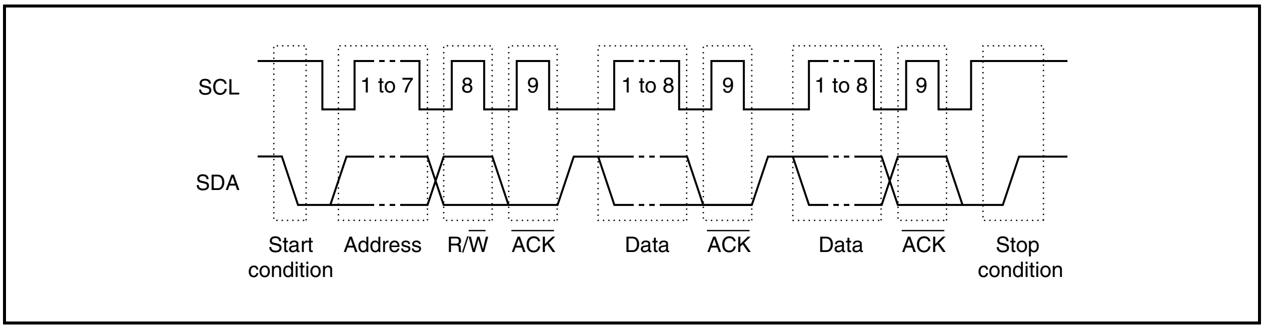
Figure 17-4. Pin Configuration Diagram



17.6 I^{2}C Bus Definitions and Control Methods

The following section describes the I^{2}C bus's serial data communication format and the status generated by the I^{2}C bus. The transfer timing for the "start condition", "address", "transfer direction specification", "data", and "stop condition" generated via the I^{2}C bus's serial data bus is shown below.

Figure 17-5. I^{2}C Bus's Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

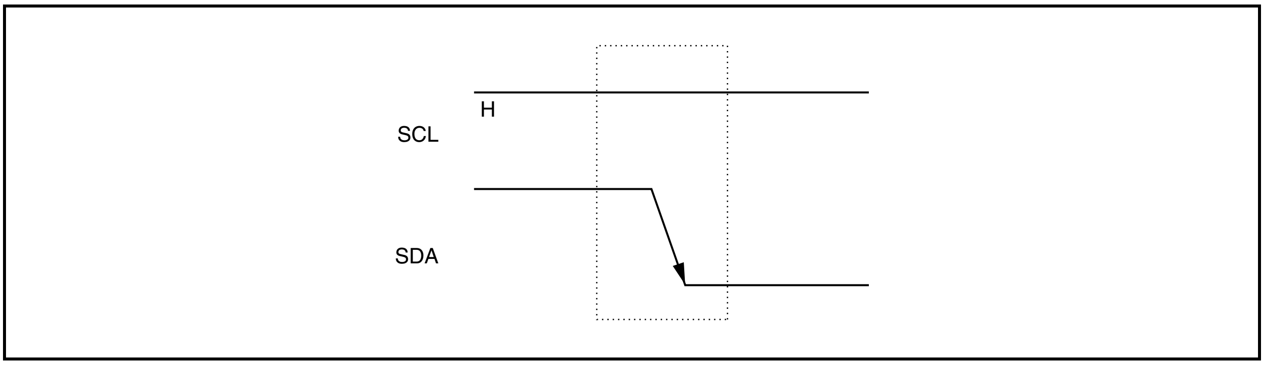
$\overline{\text{ACK}}$ can be generated by either the master or slave device (normally, it is generated by the device that receives 8-bit data).

The serial clock (SCL) is continuously output by the master device. However, in the slave device, the SCL's low-level period can be extended and a wait can be inserted.

17.6.1 Start condition

A start condition is met when the SCL pin is at high level and the SDA pin changes from high level to low level. The start conditions for the SCL pin and SDA pin are generated when the master device starts a serial transfer to the slave device. Start conditions can be detected when the device is used as a slave.

Figure 17-6. Start Conditions



A start condition is generated when the IICC0.STT0 bit is set to 1 after a stop condition has been detected (IICS0.SPD0 bit = 1). When a start condition is detected, IICS0.STD0 bit is set to 1.

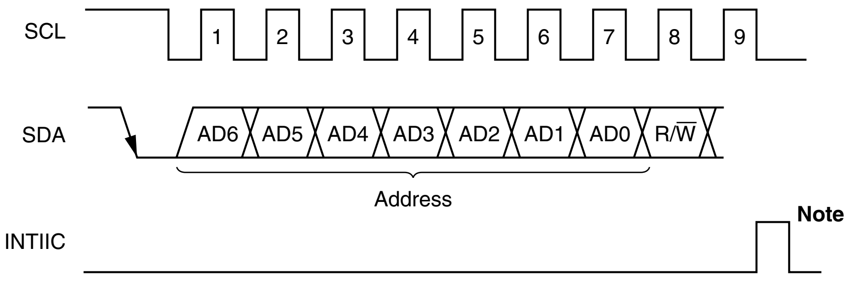
17.6.2 Addresses

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the SVA0 register. If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 17-7. Address



Note The interrupt request signal (INTIIC) is generated if a local address or extension code is received during slave device operation.

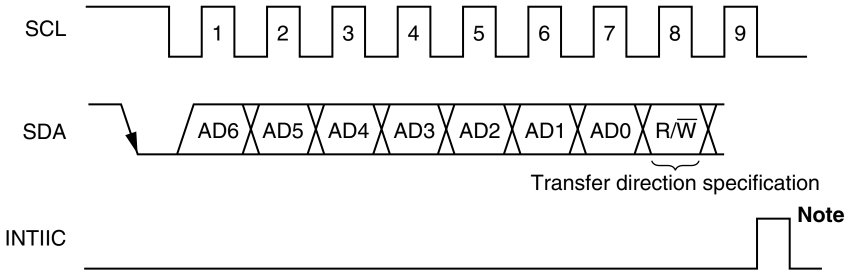
The slave address and the eighth bit, which specifies the transfer direction as described in **17.6.3 Transfer direction specification** below, are together written to the IIC0 register and are then output. Received addresses are written to the IIC0 register.

The slave address is assigned to the higher 7 bits of the IIC0 register.

17.6.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 17-8. Transfer Direction Specification



Note The interrupt request signal (INTIIC) is generated if a local address or extension code is received during slave device operation.

17.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ is used to confirm the serial data status of the transmitting and receiving devices.

The receiving device returns $\overline{\text{ACK}}$ for every 8 bits of data it receives.

The transmitting device normally receives $\overline{\text{ACK}}$ after transmitting 8 bits of data. When $\overline{\text{ACK}}$ is returned from the receiving device, the reception is judged as normal and processing continues. The detection of $\overline{\text{ACK}}$ is confirmed with the IICS0.ACKD0 bit.

When the master device is the receiving device, after receiving the final data, it does not return $\overline{\text{ACK}}$ and generates the stop condition. When the slave device is the receiving device and does not return $\overline{\text{ACK}}$, the master device generates either a stop condition or a restart condition, and then stops the current transmission. Failure to return $\overline{\text{ACK}}$ may be caused by the following factors.

(a) Reception was not performed normally.

(b) The final data was received.

(c) The receiving device (slave) does not exist for the specified address.

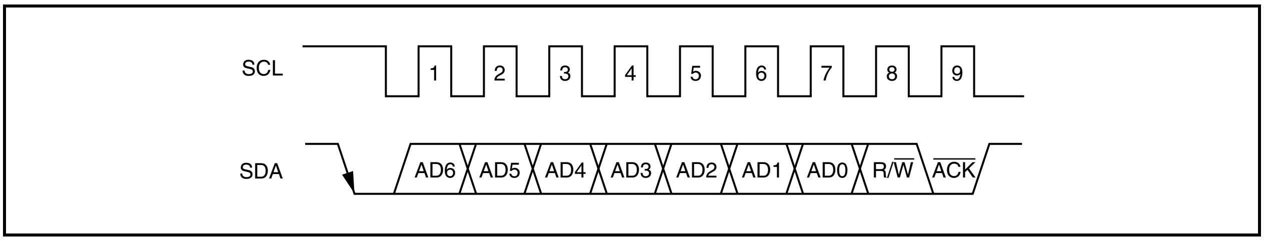
When the receiving device sets the SDA line to low level during the ninth clock, $\overline{\text{ACK}}$ is generated (normal reception).

When the IICC0.ACKE0 bit is set to 1, automatic $\overline{\text{ACK}}$ generation is enabled. Transmission of the eighth bit following the 7 address data bits causes the IICS0.TRC0 bit to be set. Normally, set the ACKE0 bit to 1 for reception (TRC0 bit = 0).

When the slave device is receiving (when TRC0 bit = 0), if the slave device cannot receive data or does not need to receive any more data, clear the ACKE0 bit to 0 to indicate to the master that no more data can be received.

Similarly, when the master device is receiving (when TRC0 bit = 0) and the subsequent data is not needed, clear the ACKE0 bit to 0 to prevent $\overline{\text{ACK}}$ from being generated. This notifies the slave device (transmitting device) of the end of the data transmission (transmission stopped).

Figure 17-9. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated regardless of the value of the ACKE0 bit. No $\overline{\text{ACK}}$ is generated if the received address is not a local address (NACK).

When receiving the extension code, set the ACKE0 bit to 1 in advance to generate $\overline{\text{ACK}}$.

The $\overline{\text{ACK}}$ generation method during data reception is based on the wait timing setting, as described by the following.

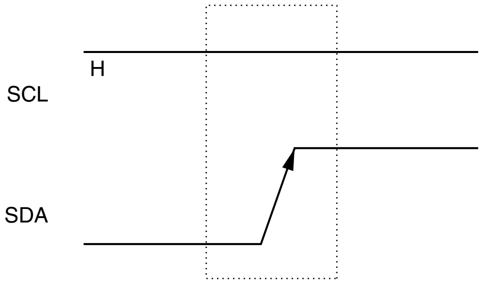
- When 8-clock wait is selected (IICC0.WTIM0 bit = 0):
  $\overline{\text{ACK}}$ is generated at the falling edge of the SCL pin's eighth clock if the ACKE0 bit is set to 1 before the wait state cancellation.
- When 9-clock wait is selected (IICC0.WTIM0 bit = 1):
  $\overline{\text{ACK}}$ is generated if the ACKE0 bit is set to 1 in advance.

17.6.5 Stop condition

When the SCL pin is at high level, changing the SDA pin from low level to high level generates a stop condition.

A stop condition is generated when serial transfer from the master device to the slave device has been completed. Stop conditions can be detected when the device is used as a slave.

Figure 17-10. Stop Condition



A stop condition is generated when the IICC0.SPT0 bit is set to 1. When the stop condition is detected, the IICS0.SPD0 bit is set to 1 and the interrupt request signal (INTIIC) is generated when the IICC0.SPIE0 bit is set to 1.

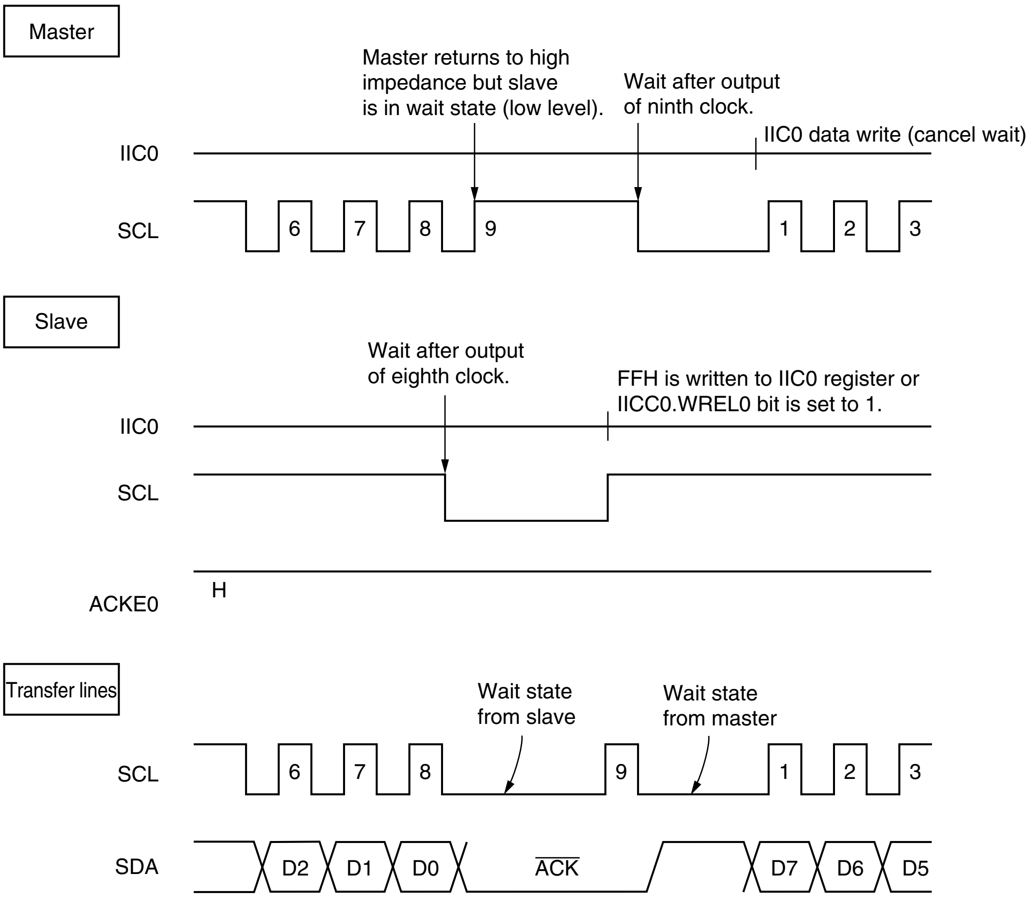
17.6.6 Wait state

The wait state is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 17-11. Wait State (1/2)

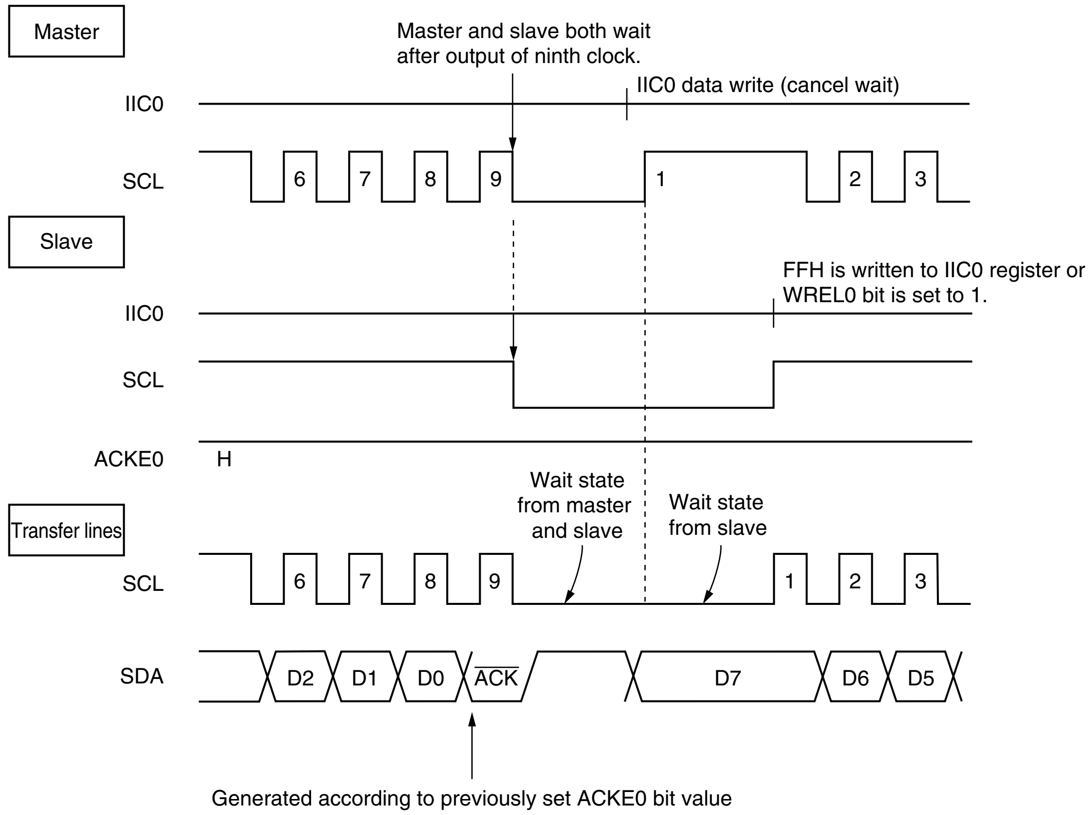
(a) When master device has a nine-clock wait and slave device has an eight-clock wait (master: transmission, slave: reception, and IICC0.ACKE0 bit = 1)



Master
Master returns to high impedance but slave is in wait state (low level).
Wait after output of ninth clock.
IIC0 data write (cancel wait)
IIC0
SCL
6
7
8
9
1
2
3
Slave
Wait after output of eighth clock.
FFH is written to IIC0 register or IICC0.WREL0 bit is set to 1.
IIC0
SCL
ACKE0
H
Transfer lines
Wait state from slave
Wait state from master
SCL
6
7
8
9
1
2
3
SDA
D2
D1
D0
ACK
D7
D6
D5

Figure 17-11. Wait State (2/2)

(b) When master and slave devices both have a nine-clock wait (master: transmission, slave: reception, and ACKE0 = 1)


Master
Master and slave both wait after output of ninth clock.
IIC0
IIC0 data write (cancel wait)
SCL
6
7
8
9
1
2
3
Slave
FFH is written to IIC0 register or WREL0 bit is set to 1.
IIC0
SCL
ACKE0
H
Transfer lines
Wait state from master and slave
Wait state from slave
SCL
6
7
8
9
1
2
3
SDA
D2
D1
D0
ACK
D7
D6
D5
Generated according to previously set ACKE0 bit value

A wait state is automatically generated after a start condition is generated. Moreover, a wait state is automatically generated depending on the setting of the IICC0.WTIM0 bit.

Normally, when the IICC0.WREL0 bit is set to 1 or when FFH is written to the IIC0 register, the wait status is canceled and the transmitting side writes data to the IIC0 register to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting the IICC0.STT0 bit to 1
- By setting the IICC0.SPT0 bit to 1

17.6.7 Wait state cancellation method

In the case of I^{2}C, wait state can be canceled normally in the following ways.

- By writing data to the IIC0 register
- By setting the IICC0.WREL0 bit to 1 (wait state cancellation)
- By setting the IICC0.STT0 bit to 1 (start condition generation)[Note]
- By setting the IICC0.SPT0 bit to 1 (stop condition generation)[Note]

Note Master only

If any of these wait state cancellation actions is performed, I^{2}C will cancel wait state and restart communication.

When canceling wait state and sending data (including address), write data to the IIC0 register.

To receive data after canceling wait state, or to end data transmission, set the WREL0 bit to 1.

To generate a restart condition after canceling wait state, set the STT0 bit to 1.

To generate a stop condition after canceling wait state, set the SPT0 bit to 1.

Execute cancellation only once for each wait state.

For example, if data is written to the IIC0 register following wait state cancellation by setting the WREL0 bit to 1, conflict between the SDA line change timing and IIC0 register write timing may result in the data output to the SDA line may be incorrect.

Even in other operations, if communication is stopped halfway, clearing the IICC0.IICE0 bit to 0 will stop communication, enabling wait state to be cancelled.

If the I^{2}C bus dead-locks due to noise, etc., setting the IICC0.LREL0 bit to 1 causes the communication operation to be exited, enabling wait state to be cancelled.

17.7 I^{2}C Interrupt Request Signals (INTIIC)

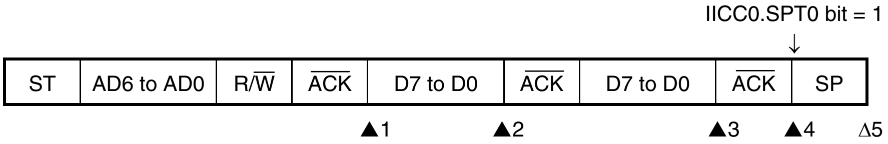
The following shows the value of the IICS0 register at the INTIIC interrupt request signal generation timing and at the INTIIC signal timing.

Remark	ST:	Start condition
	AD6 to AD0:	Address
	R/$\overline{W}$:	Transfer direction specification
	$\overline{ACK}$:	Acknowledge
	D7 to D0:	Data
	SP:	Stop condition

17.7.1 Master device operation

(1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B

▲3: IICS0 register = 1000X000B (WTIM0 bit = 1$^{\text{Note}}$)

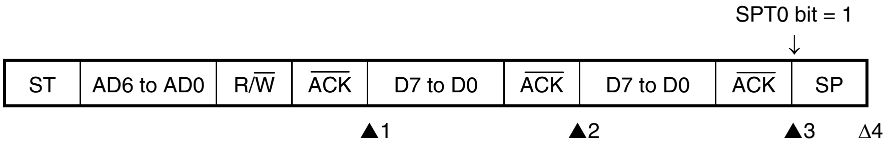
▲4: IICS0 register = 1000XX00B

△ 5: IICS0 register = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated
△: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

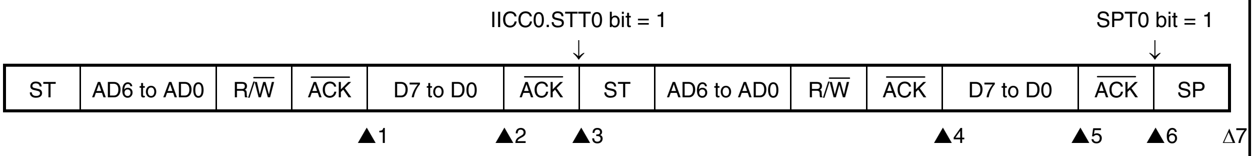
▲3: IICS0 register = 1000XX00B

△ 4: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1[Note 1])

▲3: IICS0 register = 1000XX00B (WTIM0 bit = 0[Note 2])

▲4: IICS0 register = 1000X110B

▲5: IICS0 register = 1000X000B (WTIM0 bit = 1[Note 3])

▲6: IICS0 register = 1000XX00B

△ 7: IICS0 register = 00000001B

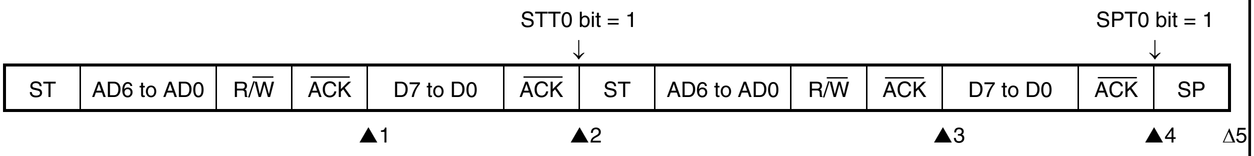
Notes 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

2. Clear the WTIM0 bit to 0 to make the settings original.

3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000XX00B

▲3: IICS0 register = 1000X110B

▲4: IICS0 register = 1000XX00B

△ 5: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

(3) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIM0 bit = 0

SPT0 bit = 1
↓

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2	▲3	▲4	Δ5

▲1: IICS0 register = 1010X110B

▲2: IICS0 register = 1010X000B

▲3: IICS0 register = 1010X000B (WTIM0 bit = 1[Note])

▲4: IICS0 register = 1010XX00B

Δ 5: IICS0 register = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

SPT0 bit = 1
↓

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2		▲3	Δ4

▲1: IICS0 register = 1010X110B

▲2: IICS0 register = 1010X100B

▲3: IICS0 register = 1010XX00B

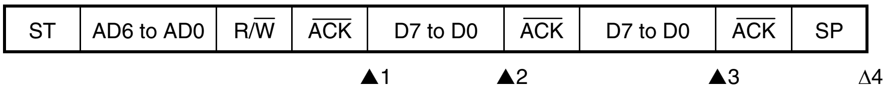
Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

17.7.2 Slave device operation (when receiving slave address data (address match))

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 0001X110B

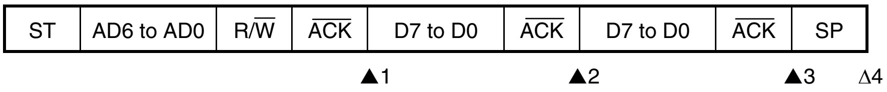
▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address match)

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
				▲1	▲2					▲3	▲4	Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

<2> When WTIM0 bit = 1 (after restart, address match)

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
				▲1		▲2				▲3		▲4 Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address mismatch (extension code))

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2				▲3		▲4		Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

<2> When WTIM0 bit = 1 (after restart, address mismatch (extension code))

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2			▲3	▲4		▲5	Δ6

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X110B

▲5: IICS0 register = 0010XX00B

Δ 6: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1	▲2					▲3			Δ4

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 00000110B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

<2> When WTIM0 bit = 1 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2				▲3			Δ4

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 00000110B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

17.7.3 Slave device operation (when receiving extension code)

Always under communication when receiving the extension code.

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICC0.WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2		▲3	Δ4

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don’t care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2		▲3		▲4 Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address match)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
		▲1		▲2					▲3	▲4		△5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001X000B

△ 5: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don’t care

<2> When WTIM0 bit = 1 (after restart, address match)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
		▲1	▲2		▲3				▲4		▲5	△6

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 0001X110B

▲5: IICS0 register = 0001XX00B

△ 6: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don’t care

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, extension code reception)

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2				▲3		▲4	Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

<2> When WTIM0 bit = 1 (after restart, extension code reception)

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2		▲3			▲4	▲5		▲6 Δ7

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 0010X010B

▲5: IICS0 register = 0010X110B

▲6: IICS0 register = 0010XX00B

Δ 7: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2					▲3		△4

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 00000110B

△ 4: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2		▲3				▲4		△5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 00000110B

△ 5: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

17.7.4 Operation without communication

(1) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
								Δ1

Δ 1: IICS0 register = 00000001B

Remark Δ: Generated only when IICC0.SPIE0 bit = 1

17.7.5 Arbitration loss operation (operation as slave after arbitration loss)

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC interrupt occurrence.

(1) When arbitration loss occurs during transmission of slave address data

<1> When IICC0.WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2		▲3		Δ4

▲1: IICS0 register = 0101X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2		▲3	Δ4

▲1: IICS0 register = 0101X110B

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) When arbitration loss occurs during transmission of extension code

<1> When WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2		▲3	Δ4

▲1: IICS0 register = 0110X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2		▲3		▲4 Δ5

▲1: IICS0 register = 0110X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

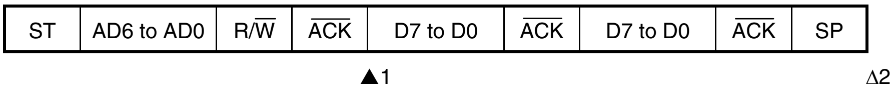
Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

17.7.6 Operation when arbitration loss occurs (no communication after arbitration loss)

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC interrupt occurrence.

(1) When arbitration loss occurs during transmission of slave address data

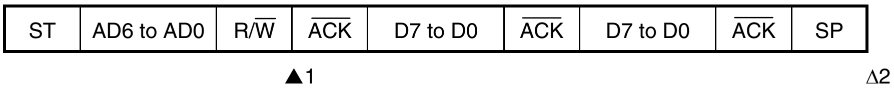


▲1: IICS0 register = 01000110B

Δ 2: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1

(2) When arbitration loss occurs during transmission of extension code



▲1: IICS0 register = 0110X010B

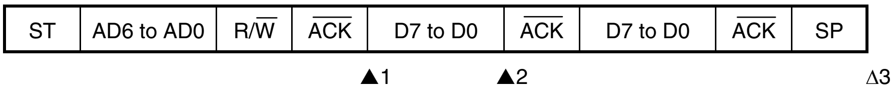
IICC0.LREL0 bit is set to 1 by software

Δ 2: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

(3) When arbitration loss occurs during data transfer

<1> When IICC0.WTIM0 bit = 0



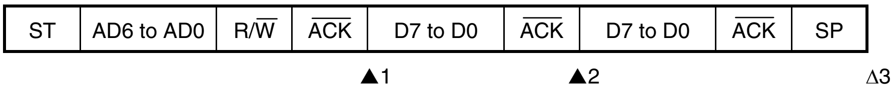
▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000000B

△ 3: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1

<2> When WTIM0 bit = 1



▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000100B

△ 3: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1

(4) When arbitration loss occurs due to restart condition during data transfer

<1> Not extension code (Example: Address mismatch)

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1					▲2			Δ3

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 01000110B

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don’t care

Dn = D6 to D0

<2> Extension code

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1				▲2				Δ3

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 0110X010B

IICC0.LREL0 bit is set to 1 by software

Δ 3: IICS0 register = 00000001B

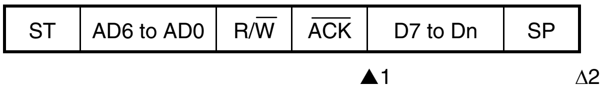
Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don’t care

Dn = D6 to D0

(5) When arbitration loss occurs due to stop condition during data transfer



▲1: IICS0 register = 1000X110B

△ 2: IICS0 register = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don’t care
Dn = D6 to D0

(6) When arbitration loss occurs due to low level of SDAn pin when attempting to generate a restart condition

<1> When WTIM0 bit = 0

IICC0.STT0 bit = 1
↓

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2	▲3	▲4				Δ5

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

▲3: IICS0 register = 1000X100B (WTIM0 bit = 0)

▲4: IICS0 register = 01000000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

<2> When WTIM0 bit = 1

IICC0.STT0 bit = 1
↓

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2		▲3			Δ4

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

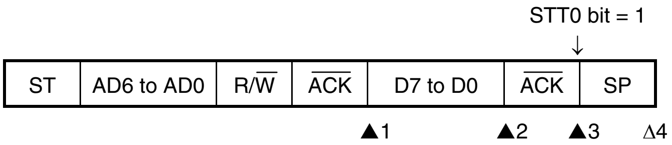
▲3: IICS0 register = 01000100B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don’t care

(7) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

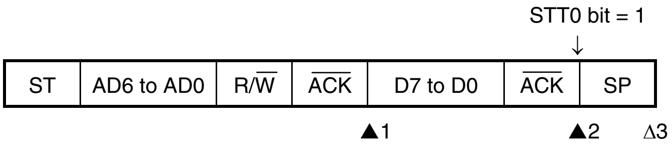
▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

▲3: IICS0 register = 1000XX00B

△ 4: IICS0 register = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

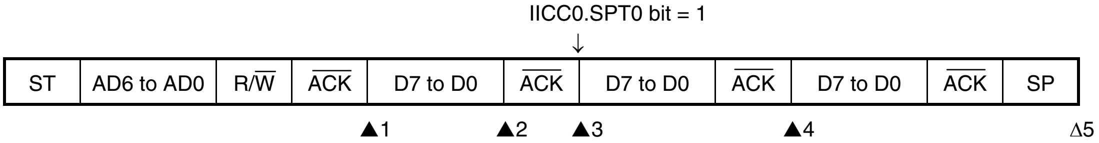
▲2: IICS0 register = 1000XX00B

△ 3: IICS0 register = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

(8) When arbitration loss occurs due to low level of SDAn pin when attempting to generate a stop condition

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

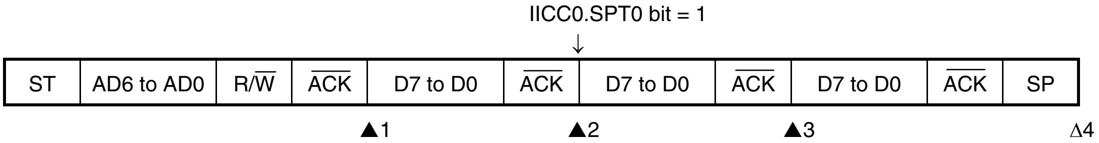
▲3: IICS0 register = 1000X100B (WTIM0 bit = 0)

▲4: IICS0 register = 01000100B

△ 5: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

▲3: IICS0 register = 01000100B

△ 4: IICS0 register = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 bit = 1
X: don't care

17.8 Interrupt Request Signal (INTIIC) Generation Timing and Wait Control

The setting of the IICC0.WTIM0 bit determines the timing by which the INTIIC signal is generated and the corresponding wait control, as shown below.

Table 17-4. INTIIC Signal Generation Timing and Wait Control

WTIM0 Bit	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9[Notes 1, 2]	8[Note 2]	8[Note 2]	9	8	8
1	9[Notes 1, 2]	9[Note 2]	9[Note 2]	9	9	9

Notes 1. The slave device's INTIIC signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the SVA0 register.

At this point, $\overline{\text{ACK}}$ is generated regardless of the value set to the IICC0.ACKE0 bit. For a slave device that has received an extension code, the INTIIC signal occurs at the falling edge of the eighth clock.

When the address does not match after restart, the INTIIC signal is generated at the falling edge of the ninth clock, but no wait occurs.

2. If the received address does not match the contents of the SVA0 register and extension codes have not been received, neither the INTIIC signal nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions in Notes 1 and 2 above regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By writing data to the IIC0 register
- By setting the IICC0.WREL0 bit (canceling wait state)
- By setting the IICC0.STT0 bit (generating start condition)[Note]
- By setting the IICC0.SPT0 bit (generating stop condition)[Note]

Note Master only

When an 8-clock wait has been selected (WTIM0 bit = 0), whether or not $\overline{\text{ACK}}$ has been generated must be determined prior to wait cancellation.

(5) Stop condition detection

The INTIIC signal is generated when a stop condition is detected.

17.9 Address Match Detection Method

When in I^{2}C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An INTIIC interrupt request signal occurs when a local address has been set to the SVA0 register and when the address set to the SVA0 register matches the slave address sent by the master device, or when an extension code has been received.

17.10 Error Detection

In I^{2}C bus mode, the status of the serial data bus (SDA) during data transmission is captured by the IIC0 register of the transmitting device, so the IIC0 register data prior to transmission can be compared with the transmitted IIC0 register data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

17.11 Extension Code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (EXC0) is set for extension code reception and an interrupt request signal (INTIIC) is issued at the falling edge of the eighth clock.
The local address stored in the SVA0 register is not affected.

(2) If 11110xx0 is set to the SVA0 register by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that the INTIIC signal occurs at the falling edge of the eighth clock.

- Higher 4 bits of data match: IICS0.EXC0 bit = 1
- 7 bits of data match: IICS0.COI0 bit = 1

(3) Since the processing after the INTIIC signal occurs differs according to the data that follows the extension code, such processing is performed by software. The slave that has received an extension code is always under communication, even if the addresses mismatch.
For example, when operation as a slave is not desired after the extension code is received, set the IICC0.LREL0 bit to 1 and the CPU will enter the next communication wait state.

Table 17-5. Bit Definitions for Major Extension Code

Slave Address	R/$\overline{W}$ Bit	Description
0000 000	0	General call address
1111 0xx	0	10-bit slave address specification (upon address authentication)
1111 0xx	1	10-bit slave address specification (upon read command issuance after address matches)

Remark For the extension codes other than above, see the I^{2}C bus specifications issued by NXP Semiconductors.

17.12 Arbitration

When several master devices simultaneously generate a start condition (when the IICC0.STT0 bit is set to 1 before the IICS0.STD0 bit is set to 1), communication among the master devices is performed as the number of clocks is adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (IICS0.ALD0 bit) is set (1) via the timing by which the arbitration loss occurred, and the SCL and SDA lines are both set for high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request signal (INTIIC) (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 bit = 1 setting that has been made by software.

For details of interrupt request timing, see **17.7 I^{2}C Interrupt Request Signals (INTIIC)**.

Figure 17-12. Arbitration Timing Example

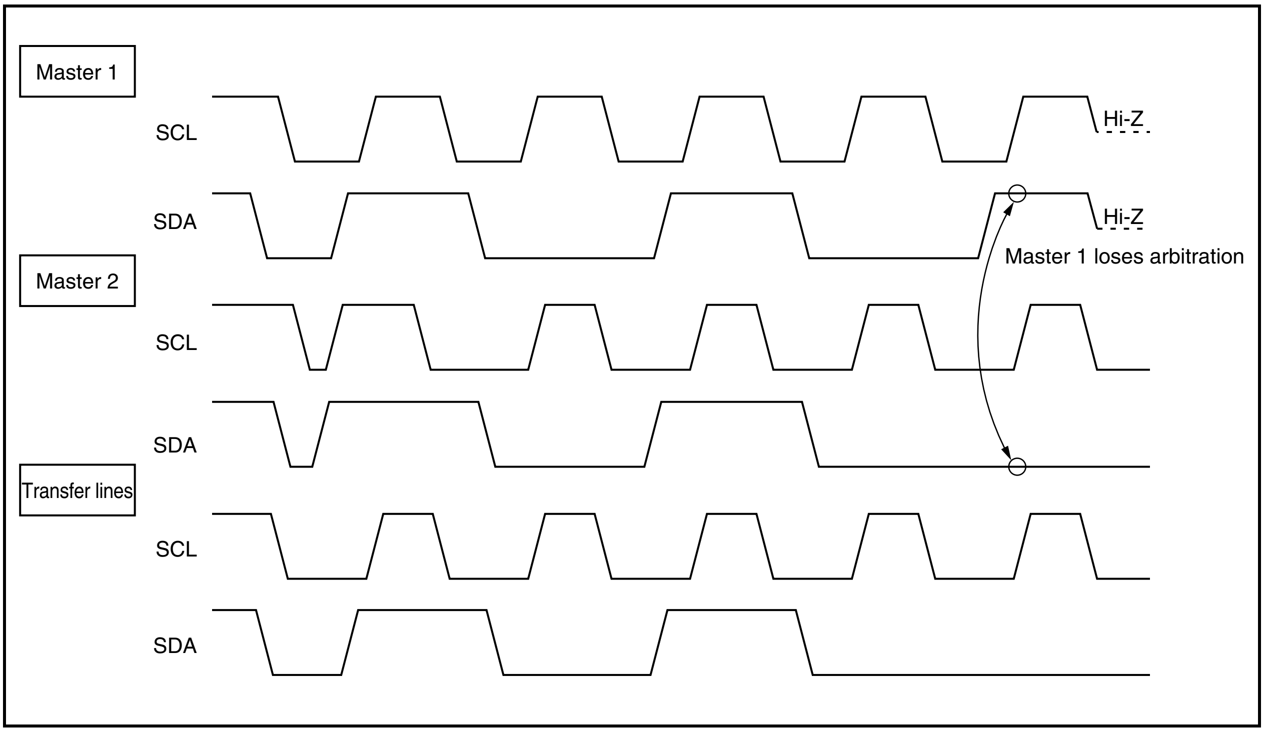


Table 17-6. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when IICC0.SPIE0 bit = 1)[Note 2]
When the SDA pin is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE0 bit = 1)[Note 2]
When the SDA pin is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When the SCL pin is at low level while attempting to generate a restart condition	

Notes 1. When the IICC0.WTIM0 bit = 1, an interrupt request occurs at the falling edge of the ninth clock. When the WTIM0 bit = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a possibility that arbitration will occur, set the SPIE0 bit = 1 for master device operation.

17.13 Wakeup Function

The I^{2}C bus slave function is a function that generates an interrupt request signal (INTIIC) when a local address or extension code has been received.

This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, the IICC0.SPIE0 bit is set regardless of the wake up function, and this determines whether interrupt requests are enabled or disabled.

17.14 Communication Reservation

17.14.1 When communication reservation function is enabled (IICF0.IICRSV0 bit = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{\text{ACK}}$ is not returned and the bus was released when the IICC0.LREL0 bit was set to “1”).

If the IICC0.STT0 bit is set (1) while the bus is not used, a start condition is automatically generated and wait status is set after the bus is released (after a stop condition is detected).

A communication is automatically started as the master by setting the IICC0.SPIE0 bit to 1, detecting the bus release due to an interrupt request (INTIIC) occurrence (detecting a stop condition), and then writing the address to the IIC0 register. Before detecting a stop condition, data written to the IIC0 register is set to invalid.

When the STT0 bit has been set (1), the operation mode (as start condition or as communication reservation) is determined according to the bus status.

If the bus has been released ....a start condition is generated
If the bus has not been released (standby mode).............communication reservation

To detect which operation mode has been determined for the STT0 bit, set the STT0 bit (1), wait for the wait period, then check the IICS0.MSTS0 bit.

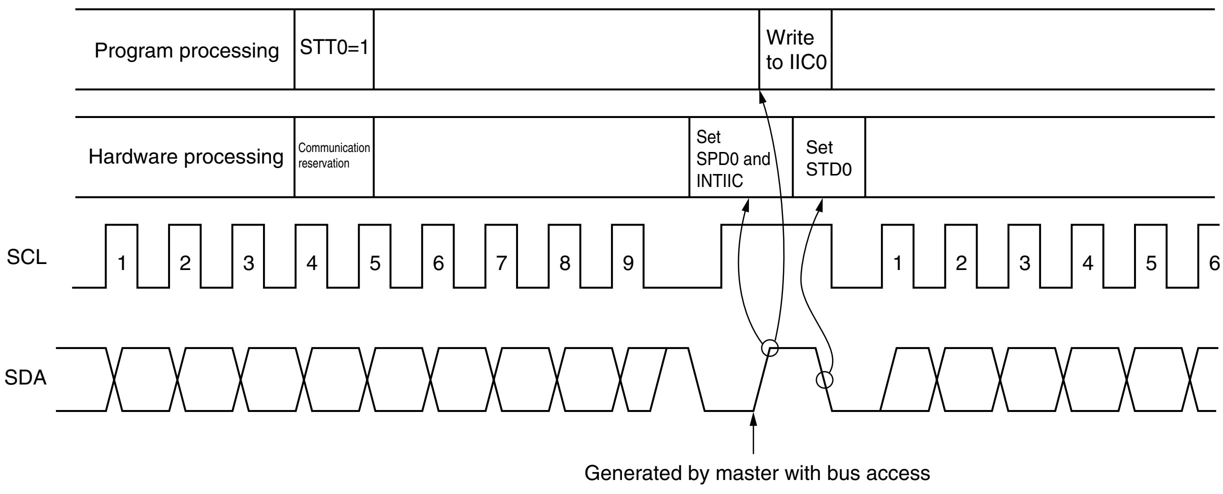
Wait periods, which should be set via software, are listed in Table 17-7. These wait periods can be set via the settings for the IICX0.CLX0, IICCL0.SMC0, and IICCL0.CL00 bits.

Table 17-7. Wait Periods

Selection Clock	CLX0	SMC0	CL00	Wait Clock	Wait Time When fxx = 100 MHz
fxx/24 (IICOCKS = 11H)	0	0	0	23 clocks	5.52 μs
fxx/32 (IICOCKS = 12H)	0	0	0	23 clocks	7.36 μs
fxx/40 (IICOCKS = 13H)	0	0	0	23 clocks	9.20 μs
fxx/16 (IICOCKS = 10H)	0	0	1	43 clocks	6.88 μs
fxx/16 (IICOCKS = 10H)	0	1	x	15 clocks	2.40 μs
fxx/24 (IICOCKS = 11H)	0	1	x	15 clocks	3.60 μs
fxx/32 (IICOCKS = 12H)	0	1	x	15 clocks	4.80 μs
fxx/24 (IICOCKS = 11H)	1	1	x	9 clocks	2.16 μs
fxx/32 (IICOCKS = 12H)	1	1	x	9 clocks	2.88 μs

The communication reservation timing is shown below.

Figure 17-13. Communication Reservation Timing



IIC0: IIC shift register 0

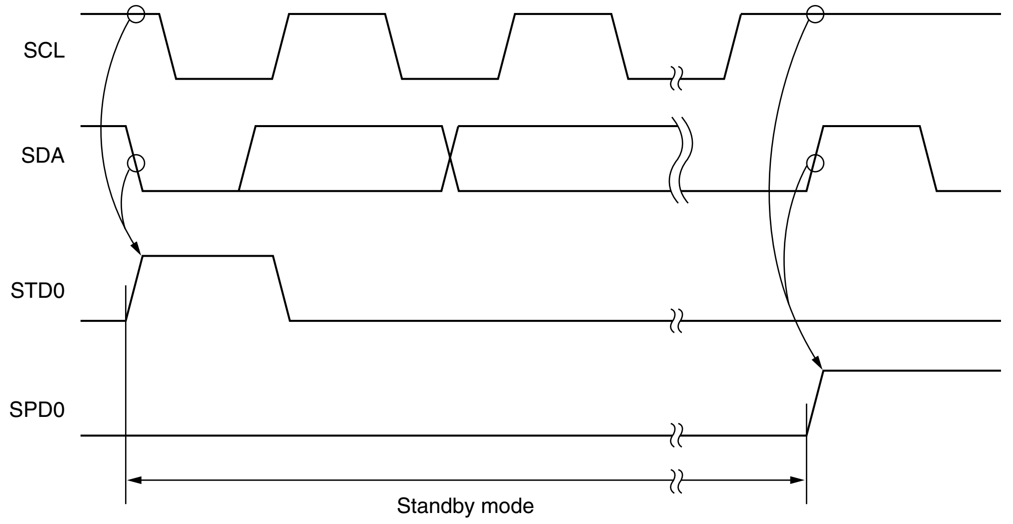
STT0: Bit 1 of IIC control register 0 (IICC0)

STD0: Bit 1 of IIC status register 0 (IICS0)

SPD0: Bit 0 of IIC status register 0 (IICS0)

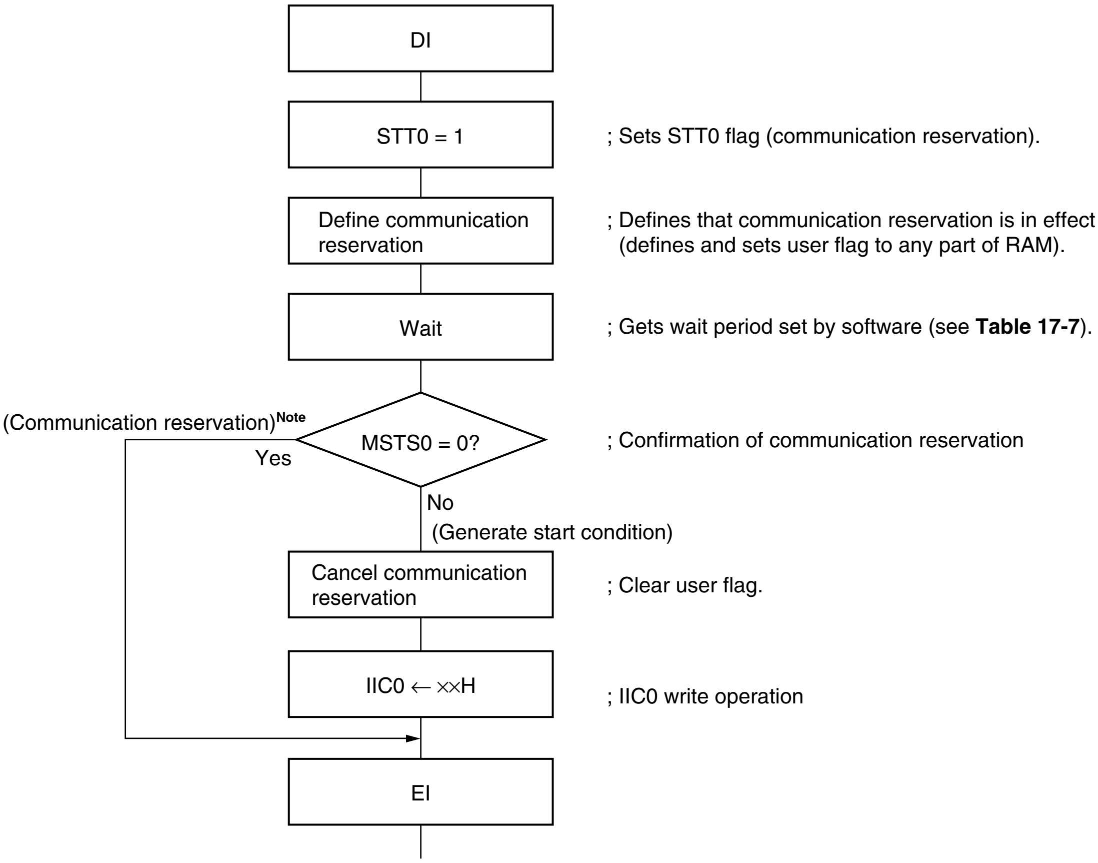
Communication reservations are accepted via the following timing. After the IICS0.STD0 bit is set to 1, a communication reservation can be made by setting the IICC0.STT0 bit to 1 before a stop condition is detected.

Figure 17-14. Timing for Accepting Communication Reservations



The communication reservation flowchart is illustrated below.

Figure 17-15. Communication Reservation Flowchart


DI
STT0 = 1
; Sets STT0 flag (communication reservation).
Define communication reservation
; Defines that communication reservation is in effect (defines and sets user flag to any part of RAM).
Wait
; Gets wait period set by software (see **Table 17-7**).
(Communication reservation)Note
MSTS0 = 0?
Yes
; Confirmation of communication reservation
No
(Generate start condition)
Cancel communication reservation
; Clear user flag.
IIC0 ← ××H
; IIC0 write operation
EI

Note The communication reservation operation executes a write to the IIC0 register when a stop condition interrupt request occurs.

17.14.2 When communication reservation function is disabled (IICF0.IICRSV0 bit = 1)

When the IICC0.STT0 bit is set when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{\text{ACK}}$ is not returned and the bus was released when the IICC0.LREL0 bit was set to 1)

To confirm whether the start condition was generated or request was rejected, check the IICF0.STCF0 flag. The time shown in Table 17-8 is required until the STCF0 flag is set after setting the STT0 bit = 1. Therefore, secure the time by software.

Table 17-8. Wait Periods

Selection Clock	CLX0	SMC0	CL00	Wait Clock	Wait Time When f_{xx} = 100 MHz
f_{xx}/24 (IICOCKS = 11H)	0	0	0	5 clocks	1.2 μs
f_{xx}/32 (IICOCKS = 12H)	0	0	0	5 clocks	1.6 μs
f_{xx}/40 (IICOCKS = 13H)	0	0	0	5 clocks	2.0 μs
f_{xx}/16 (IICOCKS = 10H)	0	0	1	5 clocks	0.8 μs
f_{xx}/16 (IICOCKS = 10H)	0	1	x	5 clocks	0.8 μs
f_{xx}/24 (IICOCKS = 11H)	0	1	x	5 clocks	1.2 μs
f_{xx}/32 (IICOCKS = 12H)	0	1	x	5 clocks	1.6 μs
f_{xx}/24 (IICOCKS = 11H)	1	1	x	5 clocks	1.2 μs
f_{xx}/32 (IICOCKS = 12H)	1	1	x	5 clocks	1.6 μs

17.15 Cautions

(1) When IICF0.STCEN0 bit = 0

Immediately after I^{2}C operation is enabled, the bus communication status (IICF0.IICBSY0 bit = 1) is recognized regardless of the actual bus status. To execute master communication in the status where a stop condition has not been detected, generate a stop condition and then release the bus before starting the master communication.

Use the following sequence for generating a stop condition.

<1> Set the IICCL0 register.
<2> Set the IICC0.IICE0 bit.
<3> Set the IICC0.SPT0 bit.

(2) When IICF0.STCEN0 bit = 1

Immediately after I^{2}C operation is enabled, the bus released status (IICBSY0 bit = 0) is recognized regardless of the actual bus status. To generate the first start condition (IICC0.STT0 bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) When the IICC0.IICE0 bit of the V850E/IG4-H and V850E/IH4-H is set to 1 while communications with other devices are in progress, the start condition may be detected depending on the status of the communication line. Be sure to set the IICC0.IICE0 bit to 1 when the SCL and SDA lines are high level.

(4) Procedure for starting or stopping I^{2}C operation

(a) Starting I^{2}C operation

<1> Select the division clock by using the IICOCKS.IICOCKS1 and IICOCKS.IICOCKS0 bits and set the IICOCKS.IICOCKSEN bit to 1 (to enable I^{2}C division clock operation).
<2> Specify the transfer speed by using the IICCL0 and IICX0 registers.
<3> Set the IICC0.IICE0 bit to 1 (to start I^{2}C operation).

When changing the transfer speed for I^{2}C, do so after clearing the IICC0.IICE0 bit to 0.

(b) Stopping I^{2}C operation

<1> Clear the IICC0.IICE0 bit to 0 (to stop I^{2}C operation).
<2> Clear the IICOCKS.IICOCKSEN bit to 0 (to disable I^{2}C division clock operation).

(5) After the IICC0.STT0 and IICC0.SPT0 bits have been set to 1, they must not be re-set without being cleared to 0 first.

(6) If transmission has been reserved, set the IICC0.SPIE0 bit to 1 so that an interrupt request is generated by the detection of a stop condition. After an interrupt request has been generated, the wait state will be released by writing communication data to I^{2}C, then transferring will begin. If an interrupt is not generated by the detection of a stop condition, transmission will halt in the wait state because an interrupt request was not generated. However, it is not necessary to set the SPIE0 bit to 1 for the software to detect the IICS0.MSTS0 bit.

17.16 Communication Operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the V850E/IG4-H and V850E/IH4-H as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I^{2}C bus multimaster system, whether the bus is released or used cannot be judged by the I^{2}C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the V850E/IG4-H and V850E/IH4-H take part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the V850E/IG4-H and V850E/IH4-H lose in arbitration and are specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

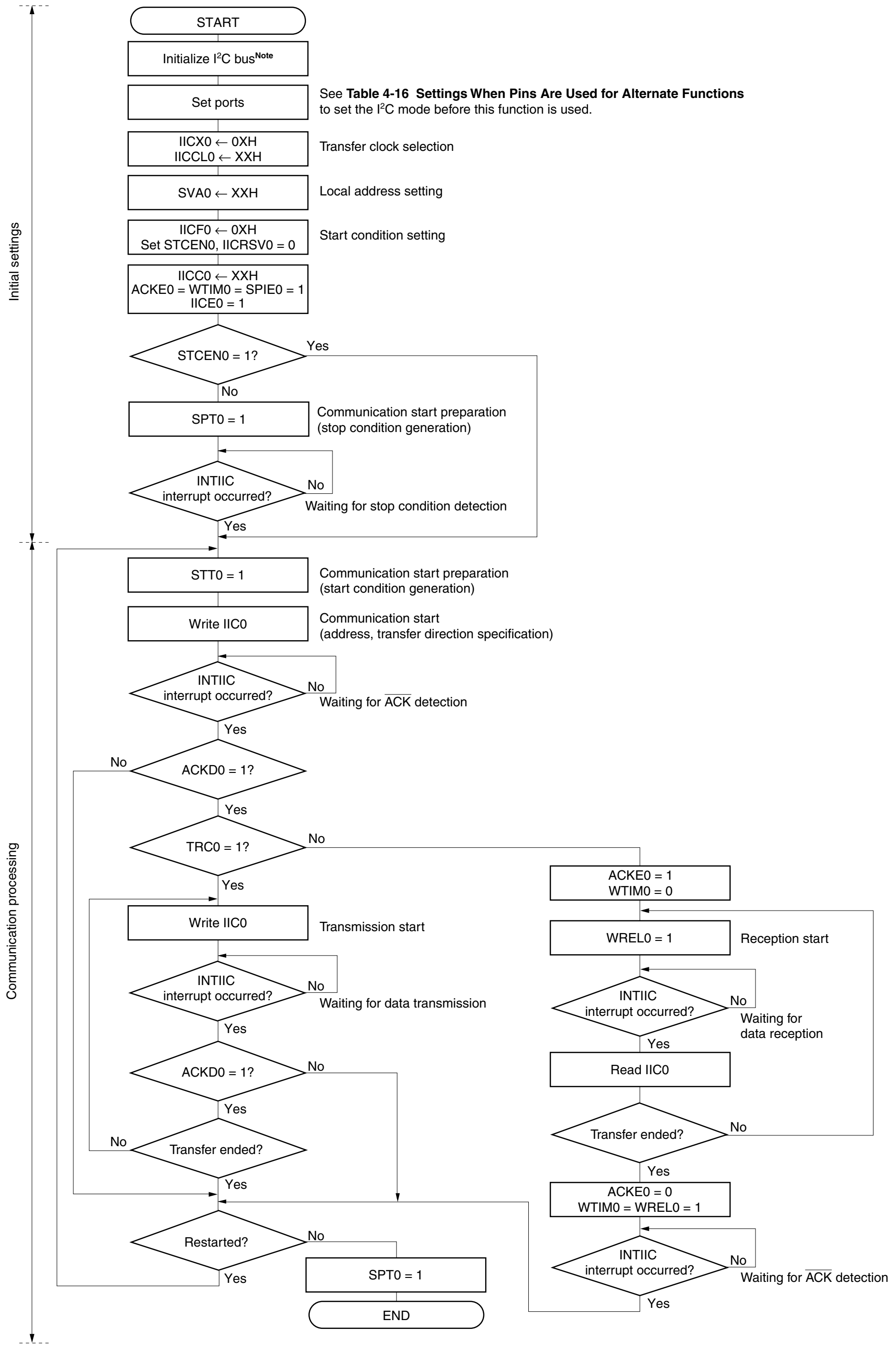
An example of when the V850E/IG4-H and V850E/IH4-H are used as the slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC interrupt occurrence (communication waiting). When the INTIIC interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

17.16.1 Master operation in single master system

Figure 17-16. Master Operation in Single Master System

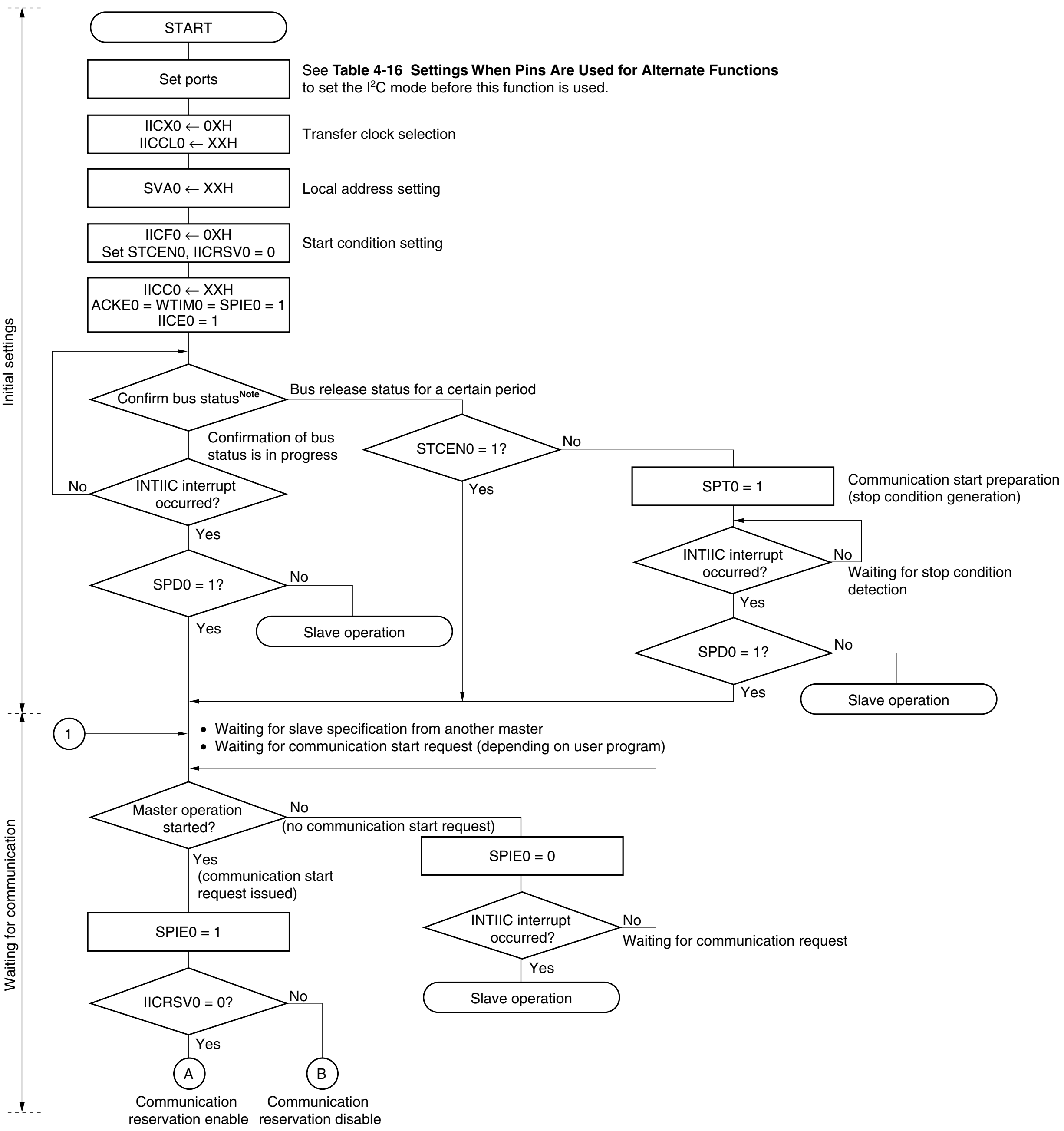

Initial settings
Communication processing
START
Initialize I²C busNote
Set ports
See Table 4-16 Settings When Pins Are Used for Alternate Functions to set the I²C mode before this function is used.
IICX0 ← 0XH
IICCL0 ← XXH
Transfer clock selection
SVA0 ← XXH
Local address setting
IICF0 ← 0XH
Set STCEN0, IICRSV0 = 0
Start condition setting
IICC0 ← XXH
ACKE0 = WTIM0 = SPIE0 = 1
IICE0 = 1
STCEN0 = 1?
Yes
No
SPT0 = 1
Communication start preparation (stop condition generation)
INTIIC interrupt occurred?
No
Waiting for stop condition detection
Yes
STT0 = 1
Communication start preparation (start condition generation)
Write IIC0
Communication start (address, transfer direction specification)
INTIIC interrupt occurred?
No
Waiting for ACK detection
Yes
ACKD0 = 1?
No
Yes
TRC0 = 1?
No
Yes
Write IIC0
Transmission start
INTIIC interrupt occurred?
No
Waiting for data transmission
Yes
ACKD0 = 1?
No
Yes
Transfer ended?
No
Yes
Restarted?
No
Yes
SPT0 = 1
END
ACKE0 = 1
WTIM0 = 0
WREL0 = 1
Reception start
INTIIC interrupt occurred?
No
Waiting for data reception
Yes
Read IIC0
Transfer ended?
No
Yes
ACKE0 = 0
WTIM0 = WREL0 = 1
INTIIC interrupt occurred?
No
Waiting for ACK detection
Yes

Note Release the I²C bus (SCL, SDA pins = high level) in conformity with the specifications of the product in communication. For example, when the EEPROM™ outputs a low level to the SDA pin, set the SCL pin to the output port and output clock pulses from that output port until when the SDA pin is constantly high level.

Remark For the transmission and reception formats, conform to the specifications of the product in communication.

17.16.2 Master operation in multimaster system

Figure 17-17. Master Operation in Multimaster System (1/3)


Initial settings
START
Set ports
See **Table 4-16 Settings When Pins Are Used for Alternate Functions** to set the I²C mode before this function is used.
IICX0 ← 0XH
IICCL0 ← XXH
Transfer clock selection
SVA0 ← XXH
Local address setting
IICF0 ← 0XH
Set STCEN0, IICRSV0 = 0
Start condition setting
IICC0 ← XXH
ACKE0 = WTIM0 = SPIE0 = 1
IICE0 = 1
Confirm bus status Note
Bus release status for a certain period
Confirmation of bus status is in progress
STCEN0 = 1?
No
Yes
SPT0 = 1
Communication start preparation (stop condition generation)
No
INTIIC interrupt occurred?
Yes
INTIIC interrupt occurred?
No
Waiting for stop condition detection
Yes
SPD0 = 1?
No
Yes
Slave operation
SPD0 = 1?
No
Yes
Slave operation
1
• Waiting for slave specification from another master
• Waiting for communication start request (depending on user program)
Waiting for communication
Master operation started?
No
(no communication start request)
Yes
(communication start request issued)
SPIE0 = 0
INTIIC interrupt occurred?
No
Waiting for communication request
Yes
Slave operation
SPIE0 = 1
IICRSV0 = 0?
No
Yes
A
B
Communication reservation enable
Communication reservation disable

Note Confirm that the bus release status (IICCL0.CLD0 bit = 1, IICCL0.DAD0 bit = 1) has been maintained for a certain period (1 frame, for example). When the SDA pin is constantly low level, determine whether to release the I^{2}C bus (SCL, SDA pins = high level) by referring to the specifications of the product in communication.

Figure 17-17. Master Operation in Multimaster System (2/3)

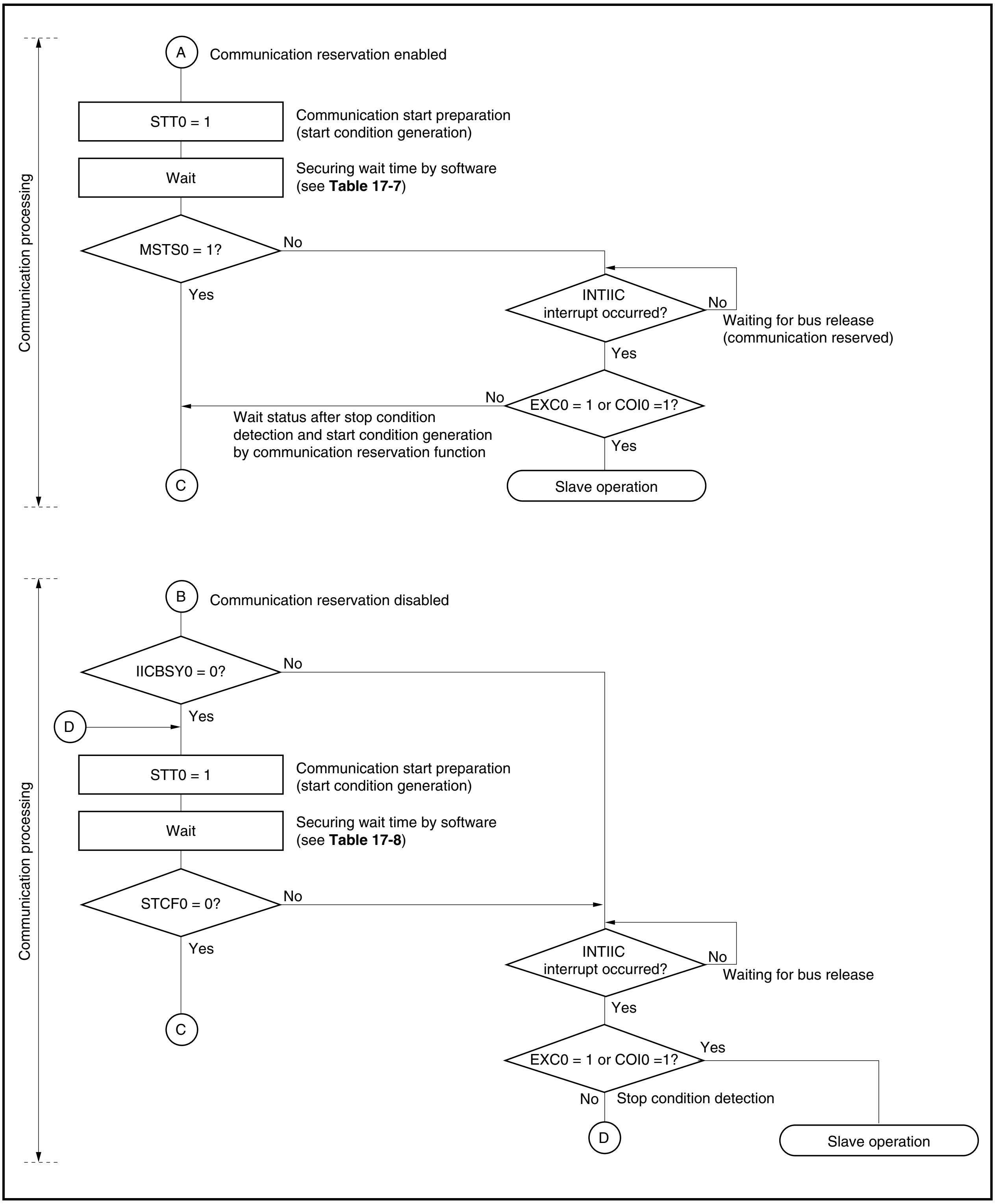
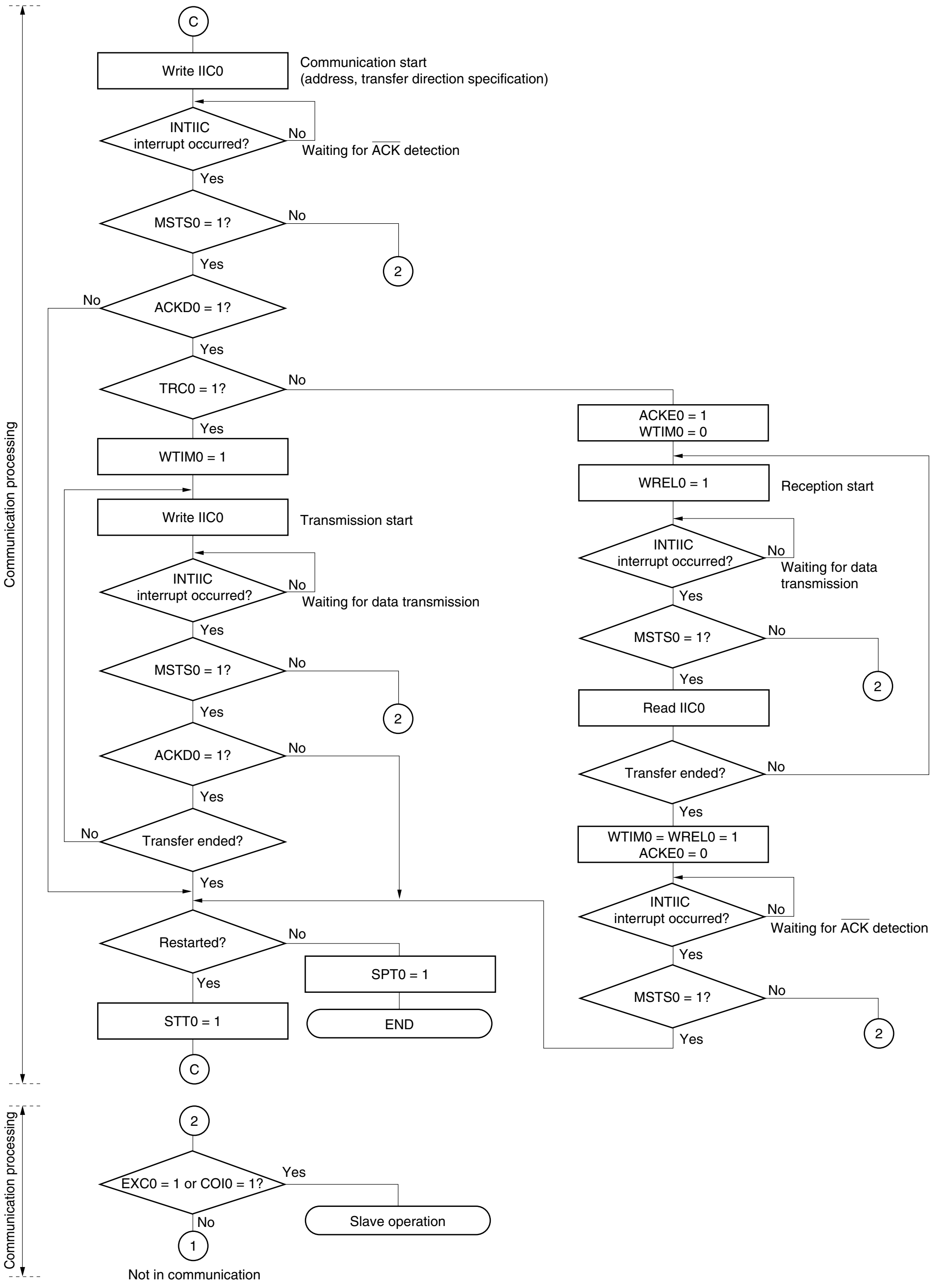

Communication processing
A
Communication reservation enabled
STT0 = 1
Communication start preparation (start condition generation)
Wait
Securing wait time by software (see Table 17-7)
MSTS0 = 1?
No
Yes
INTIIC interrupt occurred?
No
Waiting for bus release (communication reserved)
Yes
EXC0 = 1 or COI0 =1?
No
Wait status after stop condition detection and start condition generation by communication reservation function
Yes
C
Slave operation
Communication processing
B
Communication reservation disabled
IICBSY0 = 0?
No
Yes
D
STT0 = 1
Communication start preparation (start condition generation)
Wait
Securing wait time by software (see Table 17-8)
STCF0 = 0?
No
Yes
C
INTIIC interrupt occurred?
No
Waiting for bus release
Yes
EXC0 = 1 or COI0 =1?
Yes
No
Stop condition detection
D
Slave operation

Figure 17-17. Master Operation in Multimaster System (3/3)



Remarks 1. Conform the transmission and reception formats to the specifications of the product in communication.

2. When using the V850E/IG4-H and V850E/IH4-H as the master in the multimaster system, read the IICS0.MSTS0 bit for each INTIIC interrupt occurrence to confirm the arbitration result.

3. When using the V850E/IG4-H and V850E/IH4-H as the slave in the multimaster system, confirm the status using the IICS0 and IICF0 registers for each INTIIC interrupt occurrence to determine the next processing.

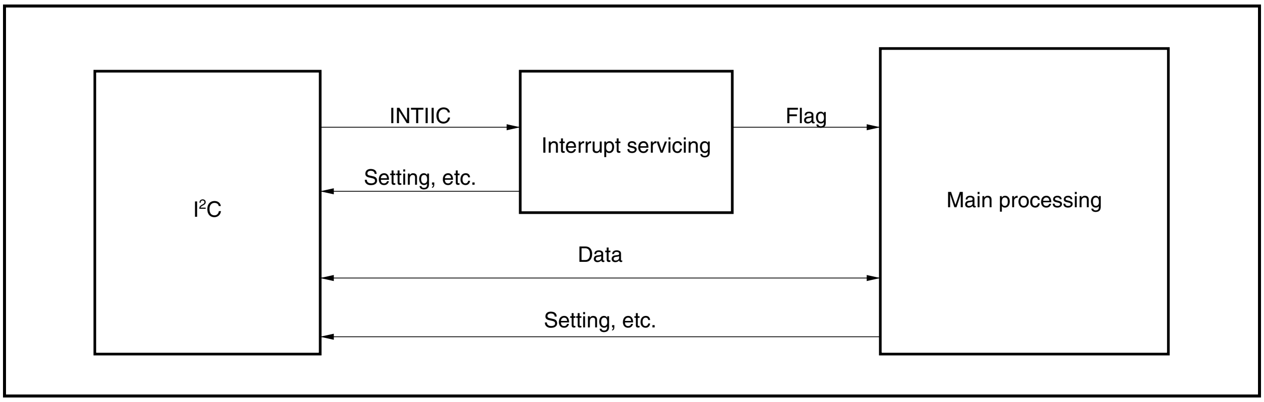
17.16.3 Slave operation

The following shows the processing procedure of the slave operation.

Basically, the operation of the slave device is event-driven. Therefore, processing by an INTIIC interrupt (processing requiring a significant change of the operation status, such as stop condition detection during communication) is necessary.

The following description assumes that data communication does not support extension codes. Also, it is assumed that the INTIIC interrupt servicing performs only status change processing and that the actual data communication is performed during the main processing.

Figure 17-18. Software Outline During Slave Operation



Therefore, the following three flags are prepared so that the data transfer processing can be performed by transmitting these flags to the main processing instead of the INTIIC signal.

(1) Communication mode flag

This flag indicates the following communication statuses.

Clear mode: Data communication not in progress

Communication mode: Data communication in progress (valid address detection stop condition detection, $\overline{\text{ACK}}$ from master not detected, address mismatch)

(2) Ready flag

This flag indicates that data communication is enabled. This is the same status as an INTIIC interrupt during normal data transfer. This flag is set in the interrupt processing block and cleared in the main processing block. The ready flag for the first data for transmission is not set in the interrupt processing block, so the first data is transmitted without clearance processing (the address match is regarded as a request for the next data).

(3) Communication direction flag

This flag indicates the direction of communication and is the same as the value of the IICS0.TRC0 bit.

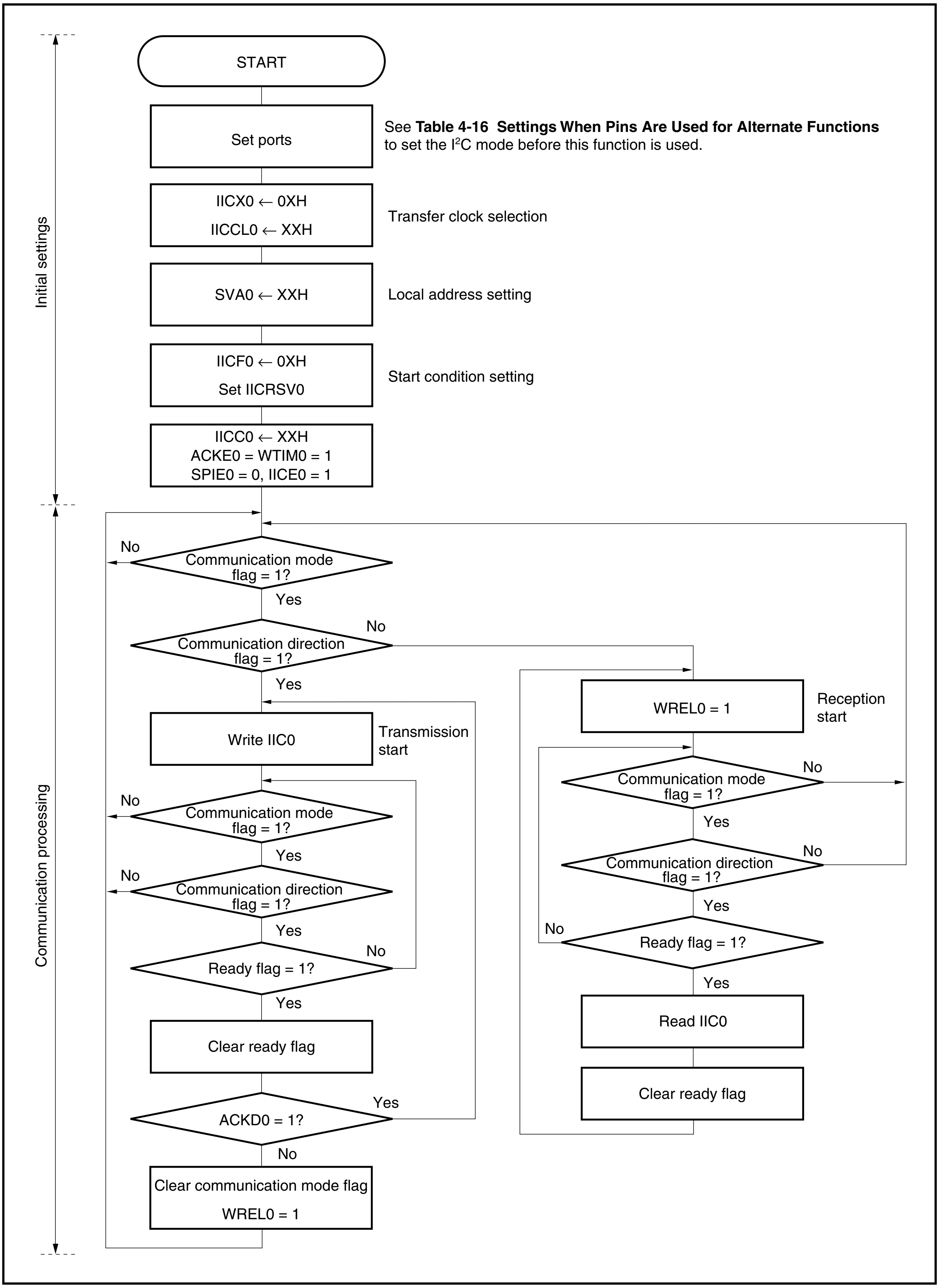
The following shows the operation of the main processing block during slave operation.

Start I^{2}C and wait for the communication enabled status. When communication is enabled, perform transfer using the communication mode flag and ready flag (the processing of the stop condition and start condition is performed by interrupts, conditions are confirmed by flags).

For transmission, repeat the transmission operation until the master device stops returning $\overline{\text{ACK}}$. When the master device stops returning $\overline{\text{ACK}}$, transfer is end.

For reception, receive the required number of data and do not return $\overline{\text{ACK}}$ for the next data immediately after transfer is end. After that, the master device generates the stop condition or restart condition. This causes exit from communications.

Figure 17-19. Slave Operation Flowchart (1)

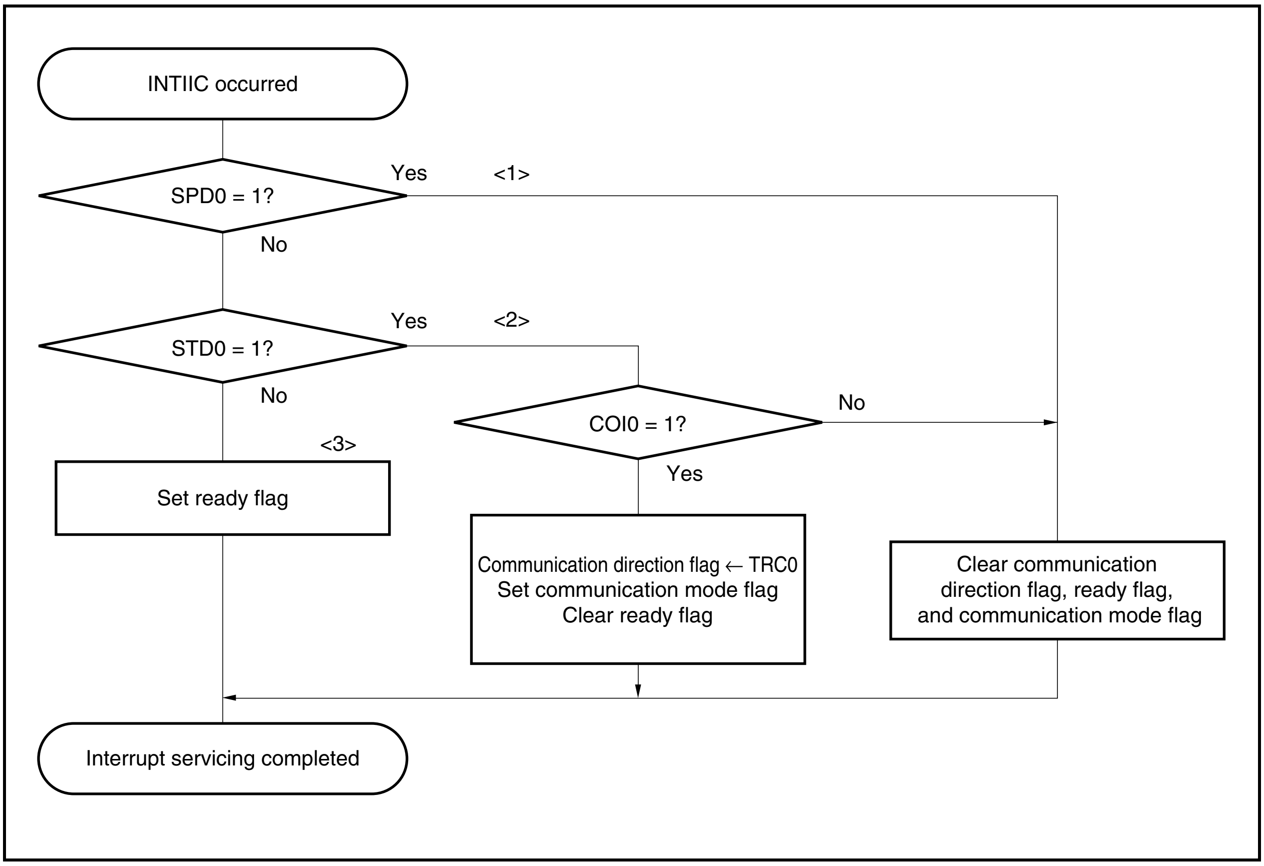


The following shows an example of the processing of the slave device by an INTIIC interrupt (it is assumed that no extension codes are used here). During an INTIIC interrupt, the status is confirmed and the following steps are executed.

<1> When a stop condition is detected, communication is terminated.
<2> When a start condition is detected, the address is confirmed. If the address does not match, communication is terminated. If the address matches, the communication mode is set and wait is released, and operation returns from the interrupt (the ready flag is cleared).
<3> For data transmission/reception, when the ready flag is set, operation returns from the interrupt while the I^{2}C bus remains in the wait status.

Remark <1> to <3> in the above correspond to <1> to <3> in **Figure 17-20 Slave Operation Flowchart (2)**.

Figure 17-20. Slave Operation Flowchart (2)


INTIIC occurred
SPD0 = 1?
Yes
<1>
No
STD0 = 1?
Yes
<2>
No
<3>
COI0 = 1?
No
Yes
Set ready flag
Communication direction flag ← TRC0
Set communication mode flag
Clear ready flag
Clear communication
direction flag, ready flag,
and communication mode flag
Interrupt servicing completed

17.17 Timing of Data Communication

When using I^{2}C bus mode, the master device generates an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the IICS0.TRC0 bit that specifies the data transfer direction and then starts serial communication with the slave device.

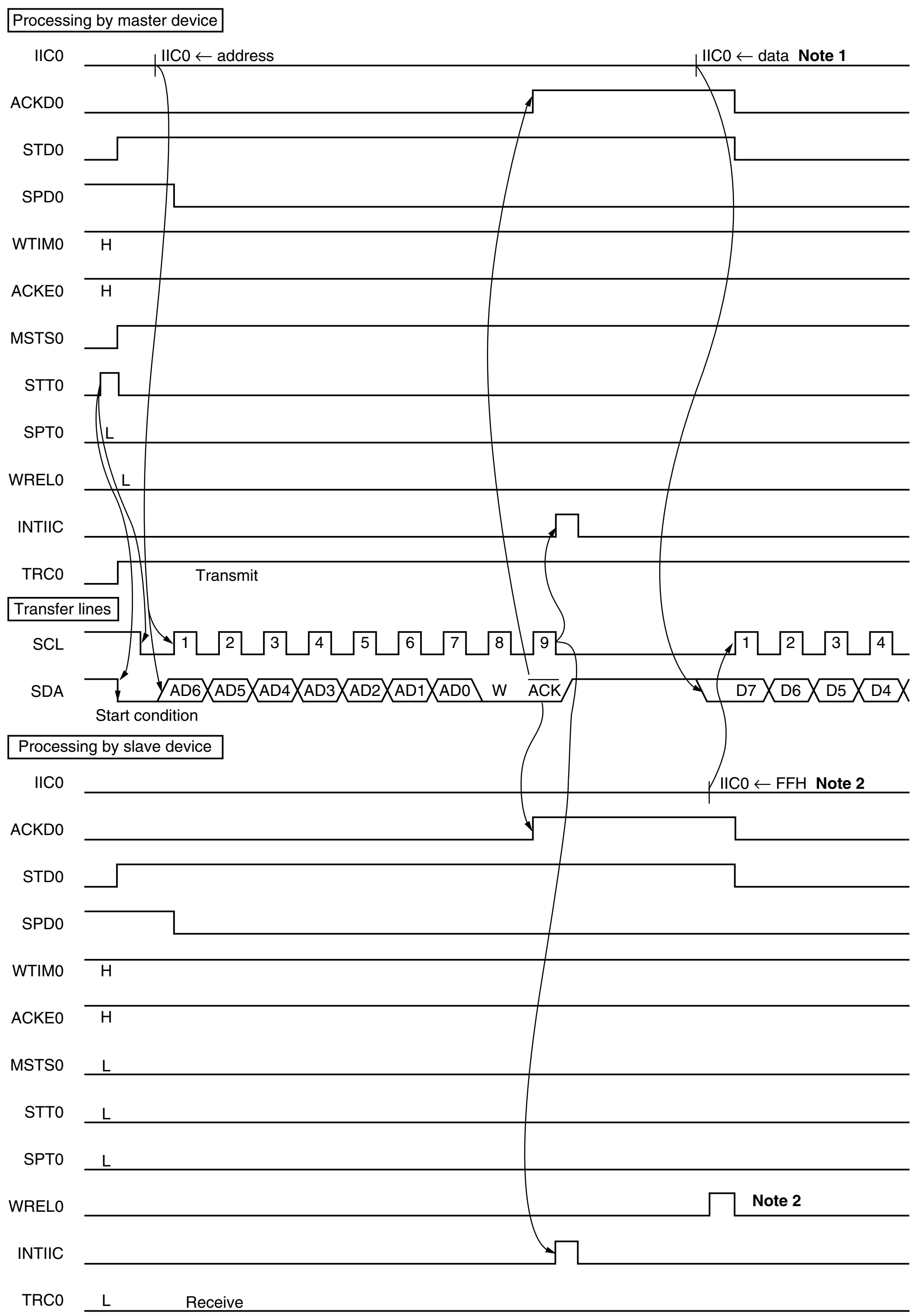
The IIC0 register's shift operation is synchronized with the falling edge of the serial clock (SCL pin). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA pin.

Data input via the SDA pin is captured by the IIC0 register at the rising edge of the SCL pin.

The data communication timing is shown below.

Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

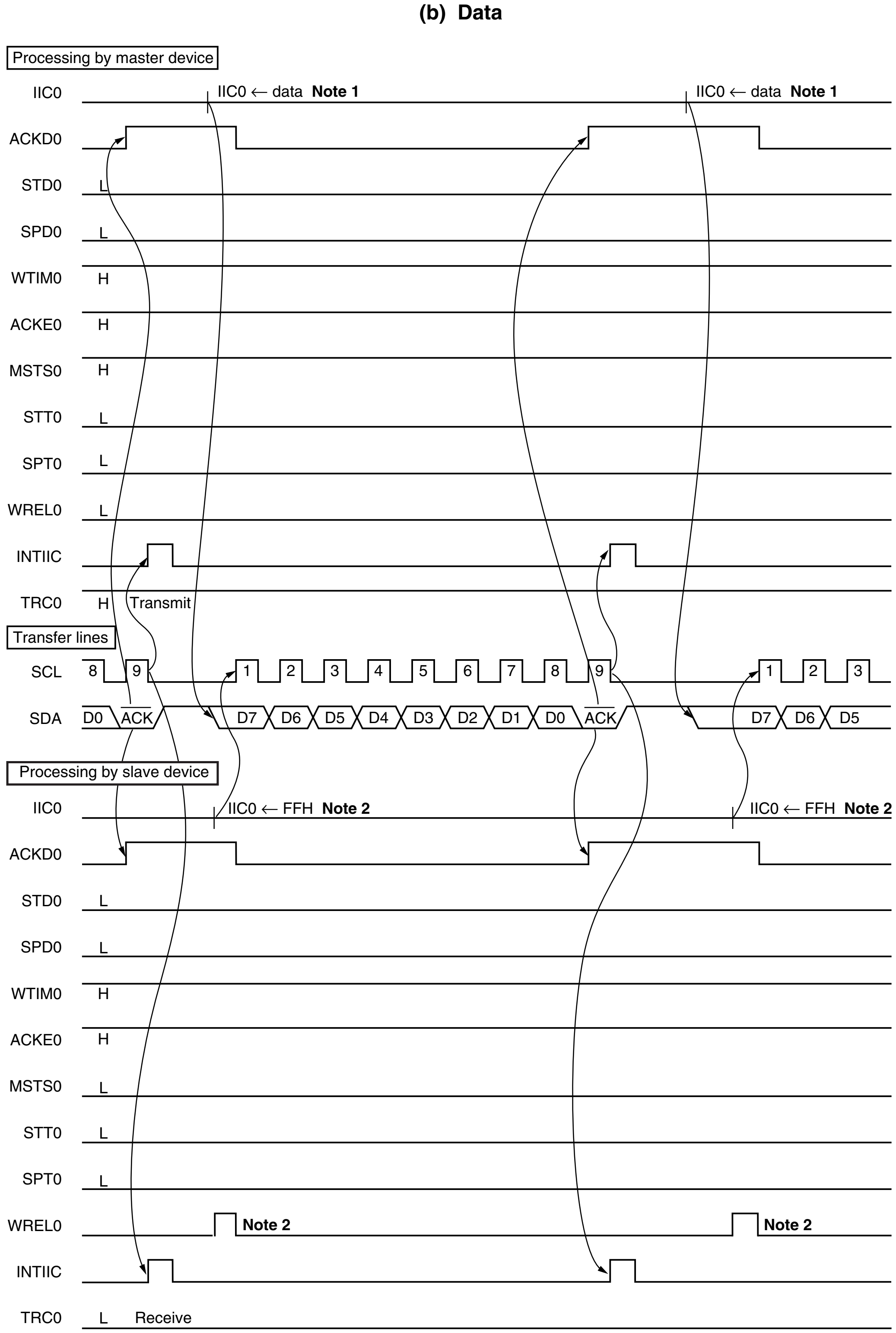
(a) Start condition ~ address



Notes 1. Cancel waits during master transmission by writing data to IIC0, not by setting WREL0.

2. To cancel slave wait, write FFH to IIC0 or set WREL0.

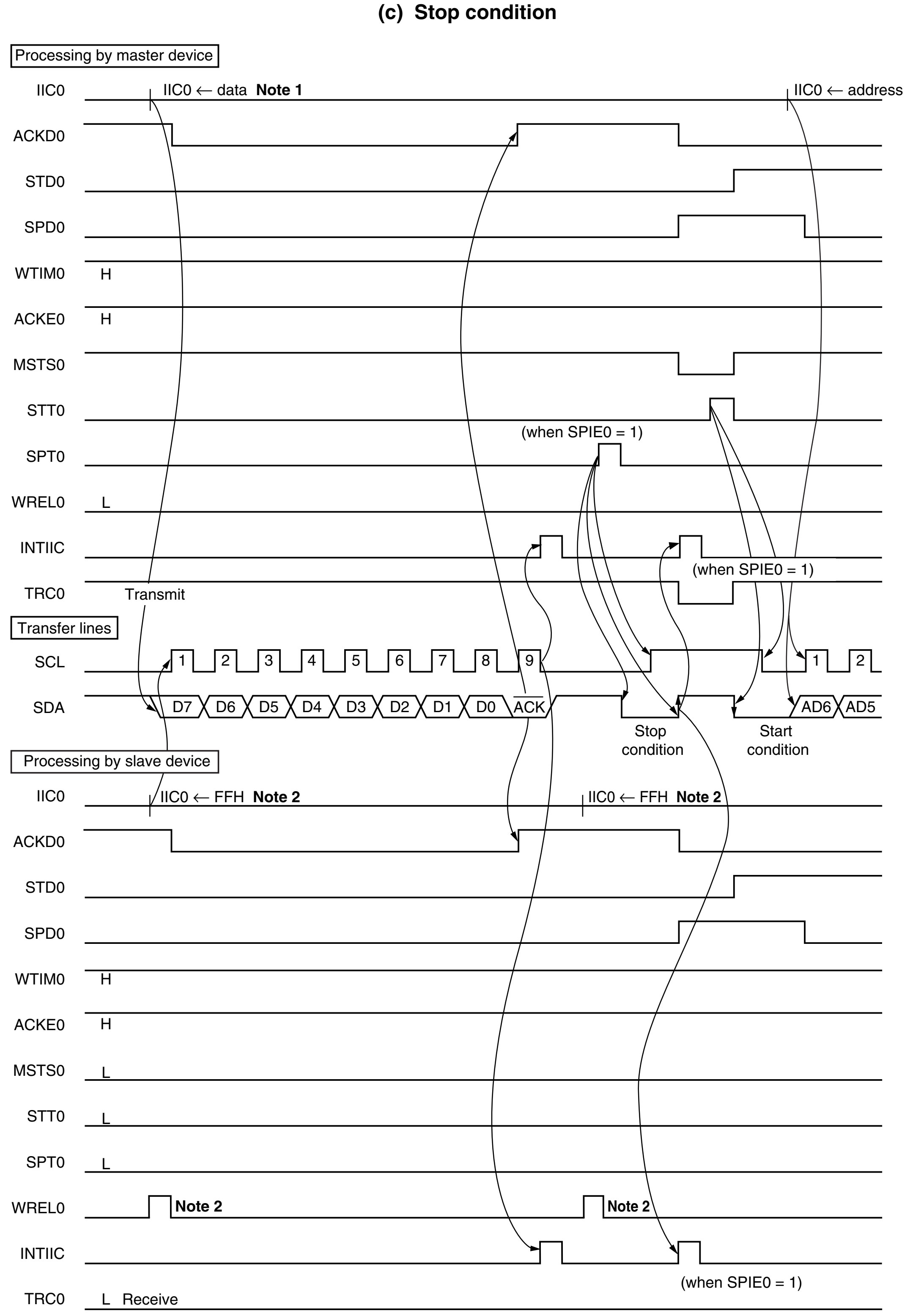
Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)



Notes 1. Cancel waits during master transmission by writing data to IIC0, not by setting WREL0.

2. To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

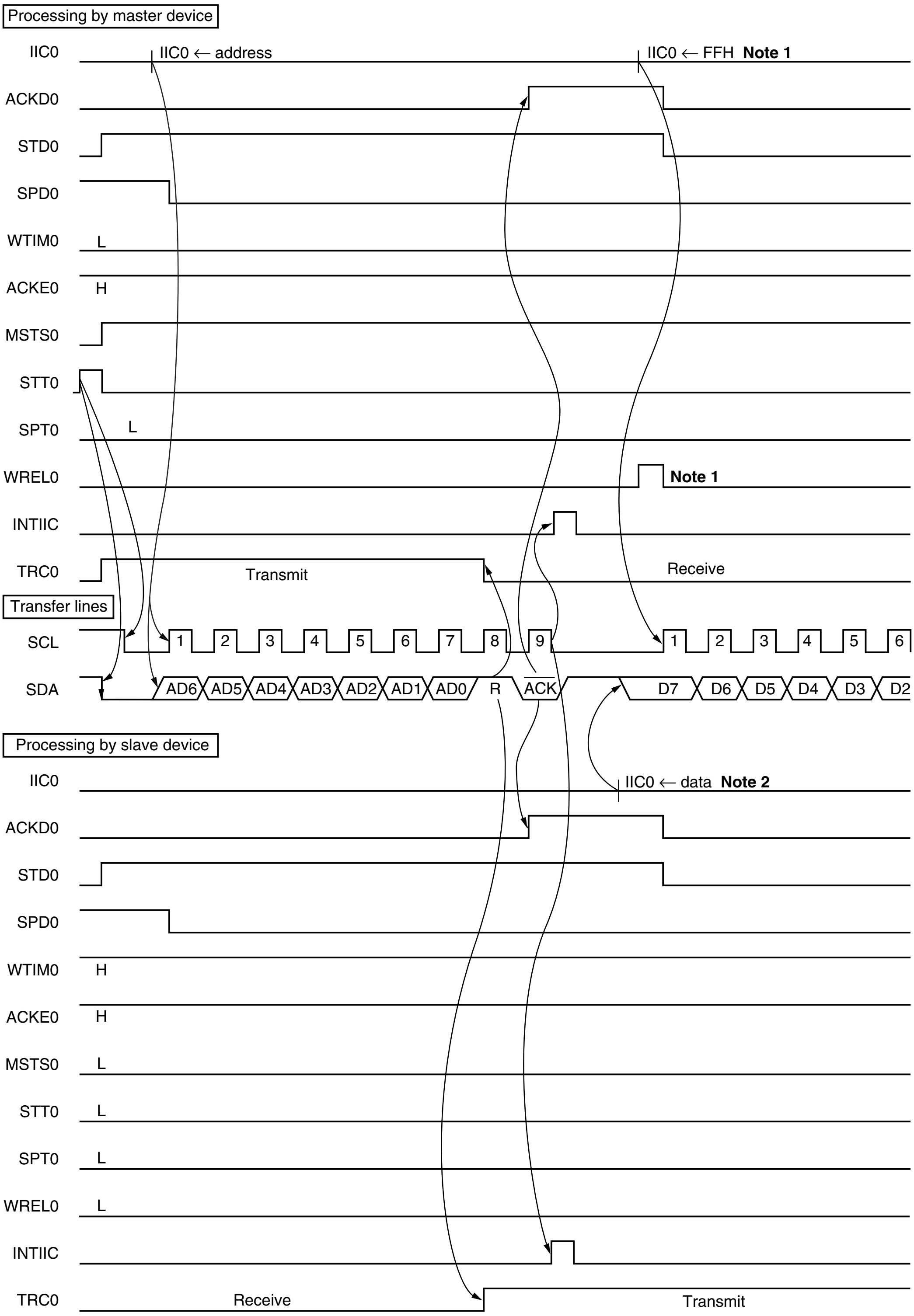


Notes 1. Cancel waits during master transmission by writing data to IIC0, not by setting WREL0.

2. To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 17-22. Example of Slave to Master Communication (When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (1/3)

(a) Start condition ~ address

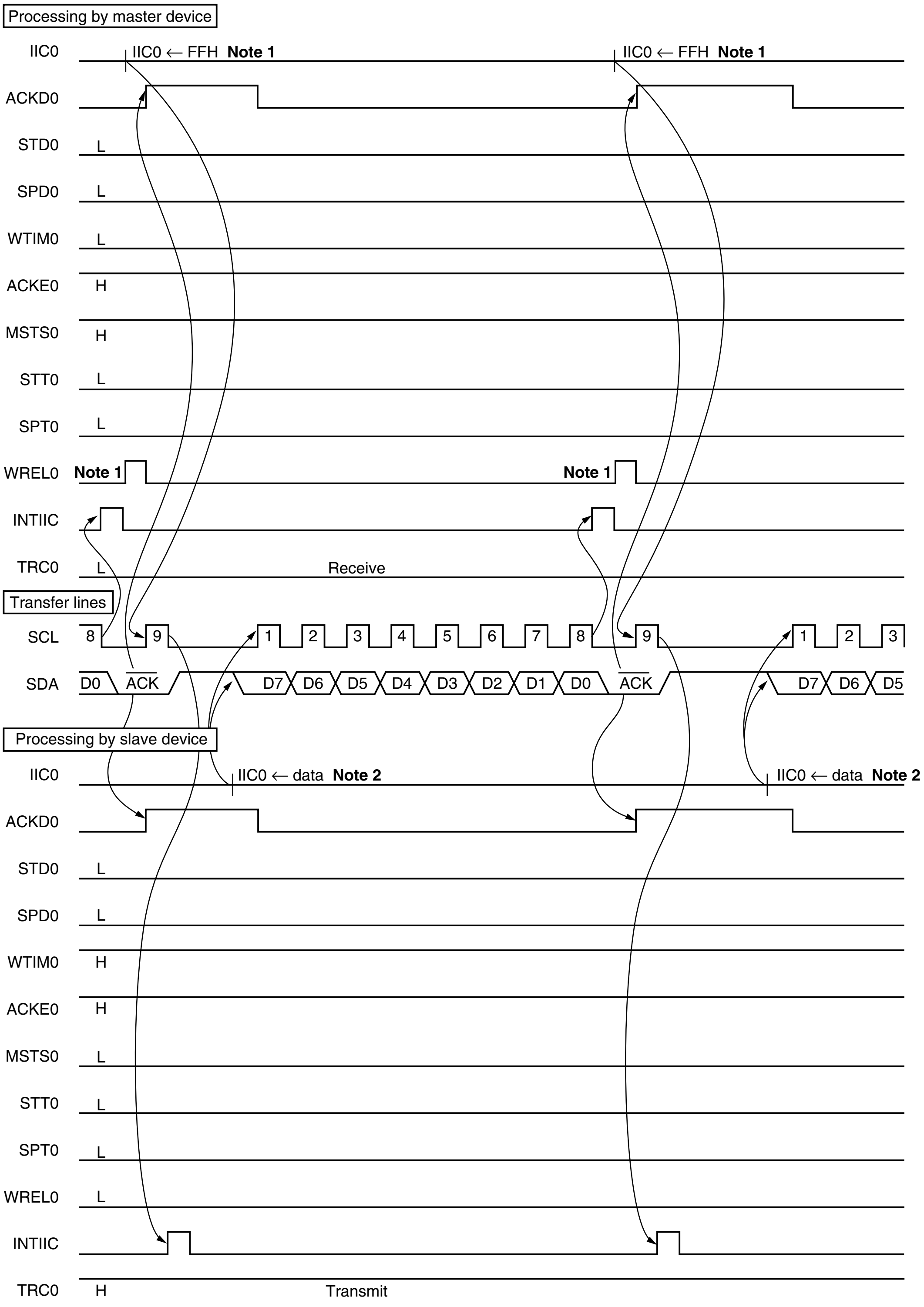


Notes 1. To cancel master wait, write FFH to IIC0 or set WREL0.

2. Cancel waits during slave transmission by writing data to IIC0, not by setting WREL0.

Figure 17-22. Example of Slave to Master Communication (When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (2/3)

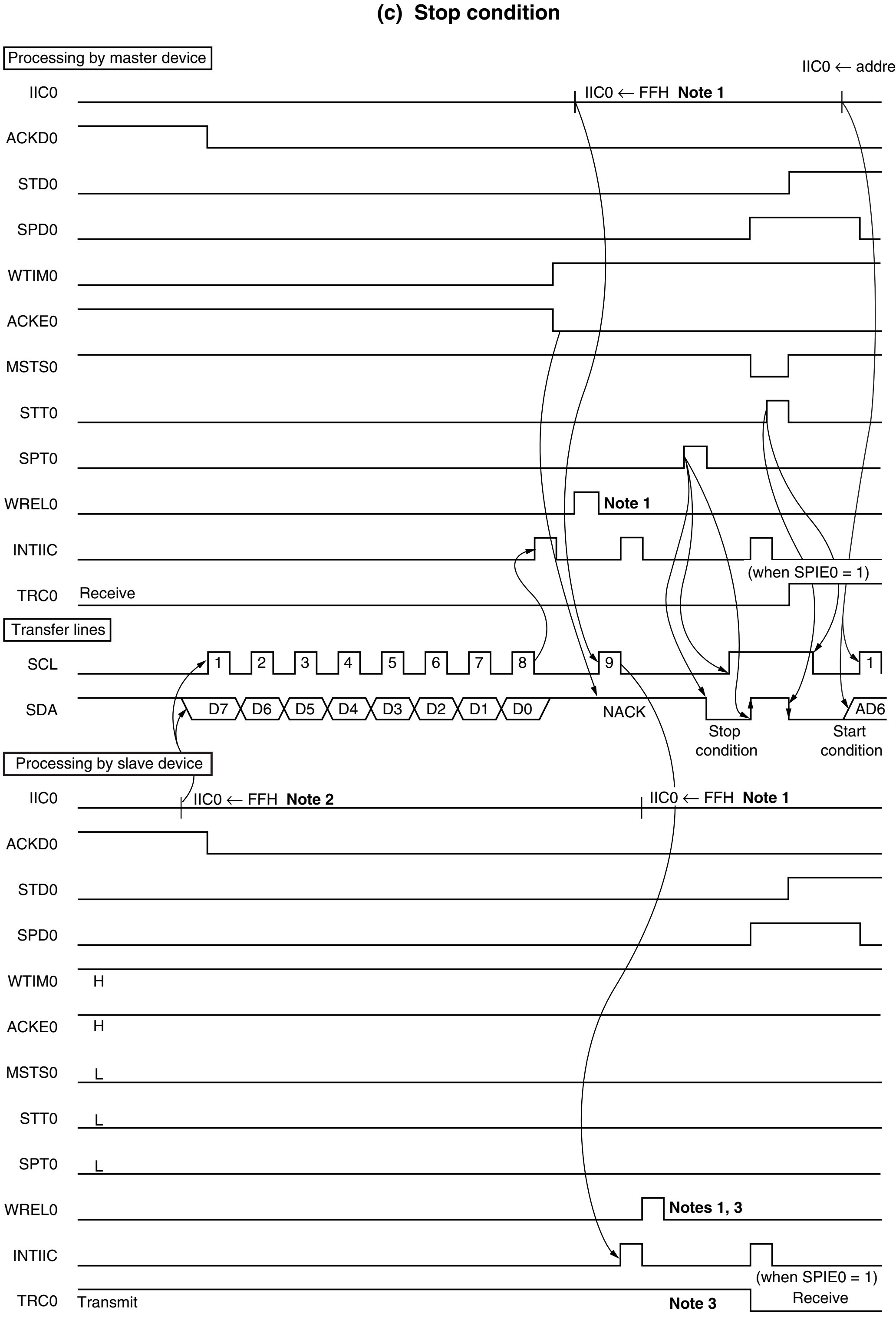
(b) Data



Notes 1. To cancel master wait, write FFH to IIC0 or set WREL0.

2. Cancel waits during slave transmission by writing data to IIC0, not by setting WREL0.

Figure 17-22. Example of Slave to Master Communication
(When 8-Clock → 9-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (3/3)



Notes 1. To cancel master wait, write FFH to IIC0 or set WREL0.

2. Cancel waits during slave transmission by writing data to IIC0, not by setting WREL0.

3. TRC0 is cleared if waits during slave transmission are canceled by setting WREL0.

CHAPTER 18 USB FUNCTION CONTROLLER (USBF)

The V850E/IG4-H and V850E/IH4-H have an internal USB function controller (USBF) conforming to the Universal Serial Bus Specification. Data communication using the polling method is performed between the USB function controller and external host device by using a token-based protocol.

18.1 Overview

- Conforms to the Universal Serial Bus Specification
- Supports 12 Mbps (full-speed) transfer
- Endpoint for transfer incorporated

Endpoint Name	FIFO Size (Bytes)	Transfer Type	Remark
Endpoint0 Read	64	Control transfer	–
Endpoint0 Write	64	Control transfer	–
Endpoint1	64 × 2	Bulk 1 transfer (IN)	2-buffer configuration
Endpoint2	64 × 2	Bulk 1 transfer (OUT)	2-buffer configuration
Endpoint3	64 × 2	Bulk 2 transfer (IN)	2-buffer configuration
Endpoint4	64 × 2	Bulk 2 transfer (OUT)	2-buffer configuration
Endpoint7	8	Interrupt transfer	–

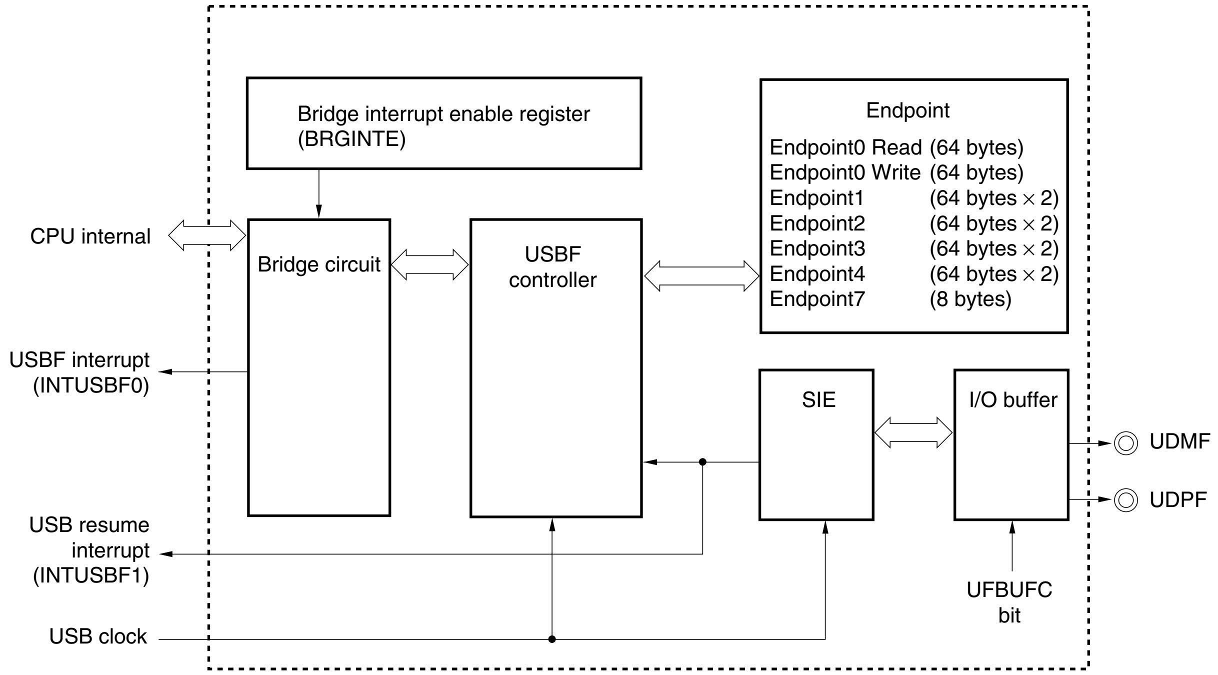
- Clock: Can be selected from internal clock (PLL output clock (96 MHz) divided by 2 (f_{USB} = 48 MHz)) or external clock (external clock input to UCLK pin (f_{USB} = 48 MHz))

Caution The group of registers described as USB function controller registers (see 18.6.2) should be accessed after a clock (the USB clock) can be supplied to the USB function controller. The USB clock can be specified as either an internal or an external clock.
If the USB function controller registers are accessed while the USB clock is not being supplied, 00H will be read in the case of a read-access. Writing is prohibited. The operation is not guaranteed if an attempt is made to write to one of the USB function controller registers while the USB clock is not being supplied.

18.2 Configuration

18.2.1 Block diagram

Figure 18-1. Block Diagram of USB Function Controller



Remarks 1. Inside broken lines: These functions are included in the USB function controller.

2. n = 0, 1

18.2.2 USB memory map

The USB function controller seen from the CPU is assigned to the CS1 space in the microcontroller. The memory space is divided for use as follows.

Table 18-1. Division of CPU Memory Space

Address	Area
00400000H to 00400092H	EPC control register area
00400100H to 00400114H	EPC data hold register area
00400144H to 004003C4H	EPC request data register area
00400400H to 00400408H	Bridge register area

18.3 External Circuit Configuration

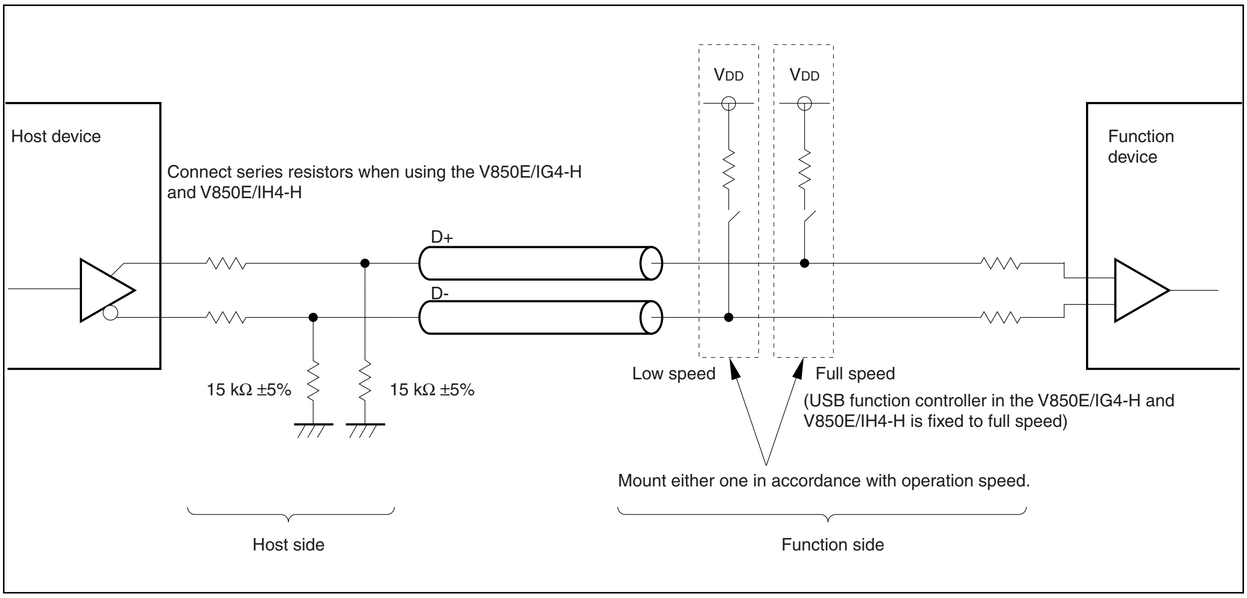
18.3.1 Outline

In USB transmission, when communication is performed with the host controller and function controller facing each other, pull-up/pull-down resistors must be connected to the USB signal (D+/D−) to identify the communication partner. Moreover in the V850E/IG4-H and V850E/IH4-H, series resistors must also be connected.

Because the V850E/IG4-H and V850E/IH4-H do not include these pull-up/pull-down resistors and series resistors, be sure to connect them externally.

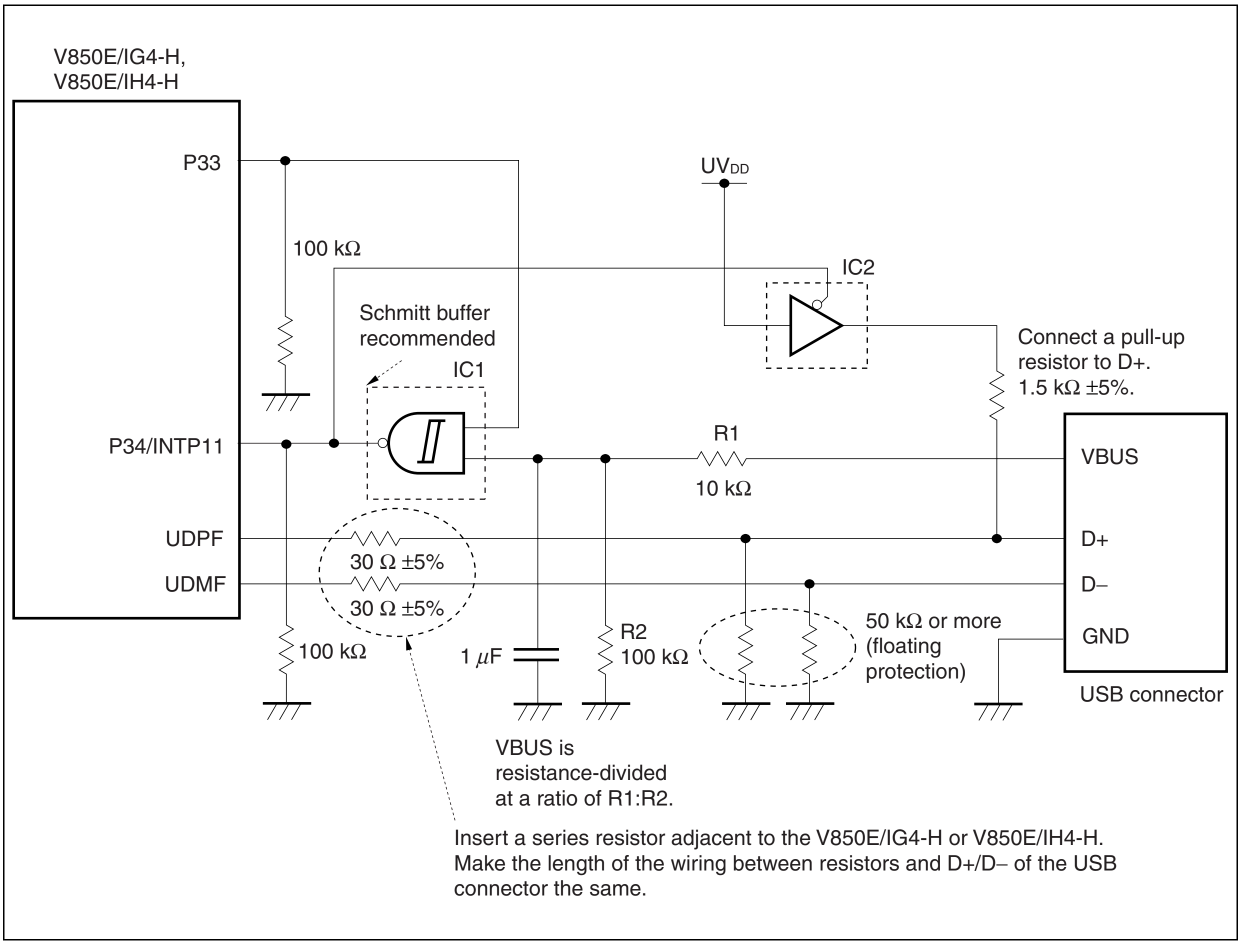
The following shows the outline configuration of the USB transmission line. For details of the external configuration, see the description provided in each section.

Figure 18-2. Outline Configuration of Pull-up, Pull-down, Series Resistors in USB Transmission Line


VDD
VDD
Host device
Function device
Connect series resistors when using the V850E/IG4-H and V850E/IH4-H
D+
D-
15 kΩ ±5%
15 kΩ ±5%
Low speed
Full speed
(USB function controller in the V850E/IG4-H and V850E/IH4-H is fixed to full speed)
Mount either one in accordance with operation speed.
Host side
Function side

18.3.2 Connection configuration

Figure 18-3. Example of USB Function Controller Connection


V850E/IG4-H,
V850E/IH4-H
P33
P34/INTP11
UDPF
UDMF
100 kΩ
UVDD
IC2
Schmitt buffer recommended
IC1
Connect a pull-up resistor to D+.
1.5 kΩ ±5%.
R1
10 kΩ
VBUS
D+
D–
GND
USB connector
30 Ω ±5%
30 Ω ±5%
100 kΩ
1 μF
R2
100 kΩ
50 kΩ or more (floating protection)
VBUS is resistance-divided at a ratio of R1:R2.
Insert a series resistor adjacent to the V850E/IG4-H or V850E/IH4-H.
Make the length of the wiring between resistors and D+/D– of the USB connector the same.

(1) Series resistor connection to D+/D–

Connect series resistors of 30 Ω ±5% to the D+/D– pins (UFDP, UFDM) of the USB function controller in the V850E/IG4-H and V850E/IH4-H. If they are not connected, the impedance rating cannot be satisfied and the output waveform may be disturbed.

Allocate the series resistors adjacent to the V850E/IG4-H or V850E/IH4-H, and make the length of the wiring between the series resistors and the USB connectors the same, to make the impedance of D+ and D– equal (a differential with 90 Ω ±5% is recommended).

(2) Pull-up control of D+

Because the function controller of the V850E/IG4-H and V850E/IH4-H is fixed to full speed (FS), be sure to pull up the D+ pin (UFDP) by 1.5 kΩ ±5% to UV_{DD}.

To disable a connection report (D+ pull up) to the USB host/HUB (such as during high priority servicing or initialization), control the pull-up resistor of D+ via a general-purpose port in the system. For a circuit such as the one shown in Figure 18-3, control the pull-up control signal and the VBUS input signal of the D+ pin by using a general-purpose port and the USB cable VBUS (AND circuit). In Figure 18-3, if the general-purpose port is high level, pulling up of D+ is prohibited.

For the IC2 in Figure 18-3, use an IC to which voltage can be applied when the system power is off.

(3) Detection of USB cable connection/disconnection

The USB function controller (USBF) requires a VBUS input signal to recognize whether the USB cable is connected or disconnected, because the state of the USBF is controlled by hardware. The voltage from the USB host or HUB (5 V) is applied as the VBUS input signal when the USB cable VBUS is connected to the USB host or HUB while the USBF power is off. Therefore, for IC1 in Figure 18-3, use an IC to which voltage can be applied when the system power is off. When disconnecting the USB cable in the circuit in Figure 18-3, the input signal to INTP11 may be unstable while the VBUS voltage is dropping. It is therefore recommended to use a Schmitt buffer for IC1 in Figure 18-3.

(4) Floating protection during initialization or when USBF is unused

When the USB function controller is initialized or unused, to avoid a floating status, pull the D+/D− pins down using a resistor of 50 kΩ or higher.

18.4 Cautions

To operate the USB function controller, the internal clock ("12 MHz external clock divided by 2 × internal clock multiplied by 8" = 48 MHz internal clock) or external clock (external clock input to UCLK pin (f_{USB} = 48 MHz)) must be used as the USB clock. When the internal clock is used as the USB clock, use a resonator with an accuracy of 12 MHz ±500 ppm (max.). When the external clock is used, supply a clock with an accuracy of 48 MHz ±500 ppm (max.) to the UCLK pin. If the USB clock accuracy drops, the transmission data cannot satisfy the USB rating.

18.5 Requests

The USB standard has a request command that reports requests from the host device to the function device to execute response processing.

The requests are received in the SETUP stage of control transfer, and most can be automatically processed via the hardware of the USB function controller (USBF).

18.5.1 Automatic requests

(1) Decode

The following tables show the request format and the correspondence between requests and decoded values.

Table 18-2. Request Format

Offset	Field Name	
0	bmRequestType	
1	bRequest	
2	wValue	Lower side
3		Higher side
4	wIndex	Lower side
5		Higher side
6	wLength	Lower side
7		Higher side

Table 18-3. Correspondence Between Requests and Decoded Values

Offset	Decoded Value								Response			Data Stage
	bmRequestType	bRequest	wValue		wIndex		wLength		Df	Ad	Cf	
Request	0	1	3	2	5	4	7	6				
GET_INTERFACE	81H	0AH	00H	00H	00H	0nH	00H	01H	STALL	STALL	ACK NAK	√
GET_CONFIGURATION	80H	08H	00H	00H	00H	00H	00H	01H	ACK NAK	ACK NAK	ACK NAK	√
GET_DESCRIPTOR Device	80H	06H	01H	00H	00H	00H	XXH	XXH[Note 1]	ACK NAK	ACK NAK	ACK NAK	√
GET_DESCRIPTOR Configuration	80H	06H	02H	00H	00H	00H	XXH	XXH[Note 1]	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS Device	80H	00H	00H	00H	00H	00H	00H	02H	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS Endpoint 0	82H	00H	00H	00H	00H	00H 80H	00H	02H	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS Endpoint X	82H	00H	00H	00H	00H	$$H	00H	02H	STALL	STALL	ACK NAK	√
CLEAR_FEATURE Device[Note 2]	00H	01H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
CLEAR_FEATURE Endpoint 0[Note 2]	02H	01H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
CLEAR_FEATURE Endpoint X[Note 2]	02H	01H	00H	00H	00H	$$H	00H	00H	STALL	STALL	ACK NAK	×
SET_FEATURE Device[Note 3]	00H	03H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_FEATURE Endpoint 0[Note 3]	02H	03H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_FEATURE Endpoint X[Note 3]	02H	03H	00H	00H	00H	$$H	00H	00H	STALL	STALL	ACK NAK	×
SET_INTERFACE	01H	0BH	00H	0#H	00H	0?H	00H	00H	STALL	STALL	ACK NAK	×
SET_CONFIGURATION[Note 4]	00H	09H	00H	00H 01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_ADDRESS	00H	05H	XXH	XXH	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×

Remark √: Data stage
×: No data stage

Notes 1. If the wLength value is lower than the prepared value, the wLength value is returned; if the wLength value is the prepared value or higher, the prepared value is returned.

2. The CLEAR_FEATURE request clears UF0 device status register L (UF0DSTL) and UF0 EPn status register L (UF0EnSL) (n = 0 to 4, 7) when ACK is received in the status stage.

Notes 3. The SET_FEATURE request sets the UF0 device status register L (UF0DSTL) and UF0 EPn status register L (UF0EnSL) (n = 0 to 4, 7) when ACK is received in the status stage. If the E0HALT bit of the UF0E0SL register is set, a STALL response is made in the status stage or data stage of control transfer for a request other than the GET_STATUS Endpoint0 request, SET_FEATURE Endpoint0 request, and a request generated by the CPUDEC interrupt request, until the CLEAR_FEATURE Endpoint0 request is received. A STALL response to an unsupported request does not set the E0HALT bit of the UF0E0SL register to 1, and the STALL response is cleared as soon as the next SETUP token has been received.

4. If the wValue is not the default value, an automatic STALL response is made.

Cautions 1. The sequence of control transfer defined by the Universal Serial Bus Specification is not satisfied under the following conditions. The operation is not guaranteed under these conditions.

- **If an IN/OUT token is suddenly received without a SETUP stage**
- **If DATA PID1 is sent in the data phase of the SETUP stage**
- **If a token of 128 addresses or more is received**
- **If the request data transmitted in the SETUP stage is of less than 8 bytes**

2. An ACK response is made even when the host transmits data other than a Null packet in the status stage.

3. If the wLength value is 00H during control transfer (read) of FW processing, a Null packet is automatically transmitted for control transfer (without data). The FW request does not automatically transmit a Null packet.

Remarks 1. Df: Default state, Ad: Addressed state, Cf: Configured state

2. n = 0 to 4
It is determined by the setting of the UF0 active interface number register (UF0AIFN) whether a request with Interface number 1 to 4 is correctly responded to, depending on whether the Interface number of the target is valid or not.

3. $$: Valid endpoint number including transfer direction
The valid endpoint is determined by the currently set Alternate Setting number (see **18.6.3 (36) UF0 active alternative setting register (UF0AAS)**, **(38) UF0 endpoint 1 interface mapping register (UF0E1IM)** to **(42) UF0 endpoint 7 interface mapping register (UF0E7IM)**).

4. ? and #: Value transmitted from host (information on Interface numbers 0 to 4)
It is determined by the UF0 active interface number register (UF0AIFN) and UF0 active alternative setting register (UF0AAS) whether an Alternate Setting request corresponding to each Interface number is correctly responded to or not, depending on whether the Interface number and Alternate Setting of the target are valid or not.

(2) Processing

The processing of an automatic request in the Default state, Addressed state, and Configured state is described below.

Remark Default state: State in which an operation is performed with the Default address
Addressed state: State after an address has been allocated
Configured state: State after SET_CONFIGURATION wValue = 1 has been correctly received

(a) CLEAR_FEATURE() request

A STALL response is made in the status stage if the CLEAR_FEATURE() request cannot be cleared, if FEATURE does not exist, or if the target is an interface or an endpoint that does not exist. A STALL response is also made if the wLength value is other than 0.

- Default state: The correct response is made when the CLEAR_FEATURE() request has been received only if the target is a device or a request for Endpoint0; otherwise a STALL response is made in the status stage.
- Addressed state: The correct response is made when the CLEAR_FEATURE() request has been received only if the target is a device or a request for Endpoint0; otherwise a STALL response is made in the status stage.
- Configured state: The correct response is made when the CLEAR_FEATURE() request has been received only if the target is a device or a request for an endpoint that exists; otherwise a STALL response is made in the status stage.

When the CLEAR_FEATURE() request has been correctly processed, the corresponding bit of the UF0 CLR request register (UF0CLR) is set to 1, the EnHALT bit of the UF0 EPn status register L (UF0EnSL) is cleared to 0, and an interrupt is issued (n = 0 to 4, 7). If the CLEAR_FEATURE() request is received when the subject is an endpoint, the toggle bit (that controls switching between DATA0 and DATA1) of the corresponding endpoint is always re-set to DATA0.

(b) GET_CONFIGURATION() request

A STALL response is made in the data stage if any of wValue, wIndex, or wLength is other than the values shown in Table 18-3.

- Default state: The value stored in the UF0 configuration register (UF0CNF) is returned when the GET_CONFIGURATION() request has been received.
- Addressed state: The value stored in the UF0CNF register is returned when the GET_CONFIGURATION() request has been received.
- Configured state: The value stored in the UF0CNF register is returned when the GET_CONFIGURATION() request has been received.

(c) GET_DESCRIPTOR() request

If the subject descriptor has a length that is a multiple of wMaxPacketSize, a Null packet is returned to indicate the end of the data stage. If the length of the descriptor at this time is less than the wLength value, the entire descriptor is returned; if the length of the descriptor is greater than the wLength value, the descriptor up to the wLength value is returned.

- Default state: The value stored in UF0 device descriptor register n (UF0DDn) and UF0 configuration/interface/endpoint descriptor register m (UF0CIEm) is returned (n = 0 to 17, m = 0 to 255) when the GET_DESCRIPTOR() request has been received.
- Addressed state: The value stored in the UF0DDn register and UF0CIEm register is returned when the GET_DESCRIPTOR() request has been received.
- Configured state: The value stored in the UF0DDn register and UF0CIEm register is returned when the GET_DESCRIPTOR() request has been received.

A descriptor of up to 256 bytes can be stored in the UF0CIEm register. To return a descriptor of more than 256 bytes, set the CDCGDST bit of the UF0MODC register to 1 and process the GET_DESCRIPTOR() request by FW.

Store the value of the total number of bytes of the descriptor set by the UF0CIEm register – 1 in the UF0 descriptor length register (UF0DSCL). The transfer data is controlled by the value of this data + 1 and wLength.

(d) GET_INTERFACE() request

If either of wValue and wLength is other than that shown in Table 18-3, or if wIndex is other than that set by the UF0 active interface number register (UF0AIFN), a STALL response is made in the data stage.

- Default state: A STALL response is made in the data stage when the GET_INTERFACE() request has been received.
- Addressed state: A STALL response is made in the data stage when the GET_INTERFACE() request has been received.
- Configured state: The value stored in the UF0 interface n register (UF0IFn) corresponding to the wIndex value is returned (n = 0 to 4) when the GET_INTERFACE() request has been received.

(e) GET_STATUS() request

A STALL response is made in the data stage if any of wValue, wIndex, or wLength is other than the values shown in Table 18-3. A STALL response is also made in the data stage if the target is an interface or an endpoint that does not exist.

- Default state: The value stored in the target status register[Note] is returned only when the GET_STATUS() request has been received and when the request is for a device or Endpoint0; otherwise a STALL response is made in the data stage.
- Addressed state: The value stored in the target status register[Note] is returned only when the GET_STATUS() request has been received and when the request is for a device or Endpoint0; otherwise a STALL response is made in the data stage.
- Configured state: The value stored in the target status register[Note] is returned only when the GET_STATUS() request has been received and when the request is for a device or an endpoint that exists; otherwise a STALL response is made in the data stage.

Note The target status register is as follows.

- If the target is a device: UF0 device status register L (UF0DSTL)
- If the target is endpoint 0: UF0 EP0 status register L (UF0E0SL)
- If the target is endpoint n: UF0 EPn status register L (UF0EnSL) (n = 1 to 4, 7)

(f) SET_ADDRESS() request

A STALL response is made in the status stage if either of wIndex or wLength is other than the values shown in Table 18-3. A STALL response is also made if the specified device address is greater than 127.

- Default state: The device enters the Addressed state and changes the USB Address value to be input to SIE into a specified address value if the specified address is other than 0 when the SET_ADDRESS() request has been received. If the specified address is 0, the device remains in the Default state.
- Addressed state: The device enters the Default state and returns the USB Address value to be input to SIE to the default address if the specified address is 0 when the SET_ADDRESS() request has been received. If the specified address is other than 0, the device remains in the Addressed state, and changes the USB Address value to be input to SIE into a specified new address value.
- Configured state: The device remains in the Configured state and returns the USB Address value to be input to SIE to the default address if the specified address is 0 when the SET_ADDRESS() request has been received. In this case, the endpoints other than endpoint 0 remain valid, and control transfer (IN), control transfer (OUT), bulk transfer and interrupt transfer for an endpoint other than endpoint 0 are also acknowledged. If the specified address is other than 0, the device remains in the Configured state and changes the USB Address value to be input to SIE into a specified new address value.

(g) SET_CONFIGURATION() request

If any of wValue, wIndex, or wLength is other than the values shown in Table 18-3, a STALL response is made in the status stage.

- Default state: The CONF bit of the UF0 mode status register (UF0MODS) and the UF0 configuration register (UF0CNF) are set to 1 if the specified configuration value is 1 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 0, the CONF bit of the UF0MODS register and UF0CNF register are cleared to 0. In other words, the device skips the Addressed state and moves to the Configured state in which it responds to the Default address.
- Addressed state: The CONF bit of the UF0MODS register and UF0CNF register are set to 1 and the device enters the Configured state if the specified configuration value is 1 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 0, the device remains in the Addressed state.
- Configured state: The CONF bit of the UF0MODS register and UF0CNF register are set to 1 and the device returns to the Addressed state if the specified configuration value is 0 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 1, the device remains in the Configured state.

If the SET_CONFIGURATION() request has been correctly processed, the target bit of the UF0 SET request register (UF0SET) is set to 1, and an interrupt is issued. All Halt Features are cleared after the SET_CONFIGURATION() request has been completed even if the specified configuration value is the same as the current configuration value. If the SET_CONFIGURATION() request has been correctly processed, the data toggle of all endpoints is always initialized to DATA0 again (it is defined that the default status, Alternative Setting 0, is set from when the SET_CONFIGURATION request is received to when the SET_INTERFACE request is received).

(h) SET_FEATURE() request

A STALL response is made in the status stage if the SET_FEATURE() request is for a Feature that cannot be set or does not exist, or if the target is an interface or an endpoint that does not exist. A STALL response is also made if the wLength value is other than 0.

- Default state: The correct response is made when the SET_FEATURE() request has been received, only if the request is for a device or Endpoint0; otherwise a STALL response is made in the status stage.
- Addressed state: The correct response is made when the SET_FEATURE() request has been received, only if the request is for a device or Endpoint0; otherwise a STALL response is made in the status stage.
- Configured state: The correct response is made when the SET_FEATURE() request has been received, only if the request is for a device or an endpoint that exists; otherwise a STALL response is made in the status stage.

When the SET_FEATURE() request has been correctly processed, the target bit of the UF0 SET request register (UF0SET) and the EnHALT bit of the UF0 EPn status register L (UF0EnSL) are set to 1, and an interrupt is issued (n = 0 to 4, 7).

(i) SET_INTERFACE() request

If wLength is other than the values shown in Table 18-3, if wIndex is other than the value set to the UF0 active interface number register (UF0AIFN), or if wValue is other than the value set to the UF0 active alternative setting register (UF0AAS), a STALL response is made in the status stage.

- Default state: A STALL response is made in the status stage when the SET_INTERFACE() request has been received.
- Addressed state: A STALL response is made in the status stage when the SET_INTERFACE() request has been received.
- Configured state: Null packet is transmitted in the status stage when the SET_INTERFACE() request has been received.

When the SET_INTERFACE() request has been correctly processed, an interrupt is issued. All the Halt Features of the endpoint linked to the target Interface are cleared after the SET_INTERFACE() request has been cleared. The data toggle of all the endpoints related to the target Interface number is always initialized again to DATA0. When the currently selected Alternative Setting is to be changed by correctly processing the SET_INTERFACE() request, the FIFO of the endpoint that is affected is completely cleared, and all the related interrupt sources are also initialized.
When the SET_INTERFACE() request has been completed, the FIFO of all the endpoints linked to the target Interface are cleared. At the same time, Halt Feature and Data PID are initialized, and the related UF0 INT status n register (UF0ISn) is cleared to 0 (n = 0 to 4). (Only Halt Feature and Data PID are cleared when the SET_CONFIGURATION request has been completed.)

18.5.2 Other requests

(1) Response and processing

The following table shows how other requests are responded to and processed.

Table 18-4. Response and Processing of Other Requests

Request	Response and Processing
GET_DESCRIPTOR String	Generation of CPUDEC interrupt request
GET_STATUS Interface	Automatic STALL response
CLEAR_FEATURE Interface	Automatic STALL response
SET_FEATURE Interface	Automatic STALL response
all SET_DESCRIPTOR	Generation of CPUDEC interrupt request
All other requests	Generation of CPUDEC interrupt request

18.6 Register Configuration

18.6.1 USB control registers

(1) USB clock select register (UCKSEL)

The UCKSEL register selects the operation clock of the USB controller.

The UCKSEL register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFF80H

	7	6	5	4	3	2	<1>	0
UCKSEL	0	0	0	0	0	0	UUSEL1	0

UUSEL1	Selection of USB function controller operation clock
0	External clock input to UCLK pin (f_{USB} = 48 MHz)
1	PLL output clock (96 MHz) divided by 2 (f_{USB} = 48 MHz)

Caution Be sure to set bits 0 and 2 to 7 to “0”.

(2) USB function control register (UFCTL)

The UFCKMSK register controls enable/disable of USB function controller operation.

The UFCKMSK register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 03H.

After reset: 03H R/W Address: FFFFFF81H

	7	6	5	4	3	2	<1>	<0>
UFCTL	0	0	0	0	0	0	UFBUFC	UFC

UFBUFC	UFC	USB function controller operation enable/disable
0	0	Operation enabled
1	1	Operation disabled
Other than above		Setting prohibited

18.6.2 USB function controller register list

(1) EPC control register

(1/2)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400000H	UF0 EP0NAK register	UF0E0N	R/W		√		00H
00400002H	UF0 EP0NAKALL register	UF0E0NA	R/W		√		00H
00400004H	UF0 EPNAK register	UF0EN	R/W		√		00H
00400006H	UF0 EPNAK mask register	UF0ENM	R/W		√		00H
00400008H	UF0 SNDSIE register	UF0SDS	R/W		√		00H
0040000AH	UF0 CLR request register	UF0CLR	R		√		00H
0040000CH	UF0 SET request register	UF0SET	R		√		00H
0040000EH	UF0 EP status 0 register	UF0EPS0	R		√		00H
00400010H	UF0 EP status 1 register	UF0EPS1	R		√		00H
00400012H	UF0 EP status 2 register	UF0EPS2	R		√		00H
00400020H	UF0 INT status 0 register	UF0IS0	R		√		00H
00400022H	UF0 INT status 1 register	UF0IS1	R		√		00H
00400024H	UF0 INT status 2 register	UF0IS2	R		√		00H
00400026H	UF0 INT status 3 register	UF0IS3	R		√		00H
00400028H	UF0 INT status 4 register	UF0IS4	R		√		00H
0040002EH	UF0 INT mask 0 register	UF0IM0	R/W		√		00H
00400030H	UF0 INT mask 1 register	UF0IM1	R/W		√		00H
00400032H	UF0 INT mask 2 register	UF0IM2	R/W		√		00H
00400034H	UF0 INT mask 3 register	UF0IM3	R/W		√		00H
00400036H	UF0 INT mask 4 register	UF0IM4	R/W		√		00H
0040003CH	UF0 INT clear 0 register	UF0IC0	W		√		FFH
0040003EH	UF0 INT clear 1 register	UF0IC1	W		√		FFH
00400040H	UF0 INT clear 2 register	UF0IC2	W		√		FFH
00400042H	UF0 INT clear 3 register	UF0IC3	W		√		FFH
00400044H	UF0 INT clear 4 register	UF0IC4	W		√		FFH
00400060H	UF0 FIFO clear 0 register	UF0FIC0	W		√		00H
00400062H	UF0 FIFO clear 1 register	UF0FIC1	W		√		00H
0040006AH	UF0 data end register	UF0DEND	R/W		√		00H
0040006EH	UF0 GPR register	UF0GPR	W		√		00H
00400074H	UF0 mode control register	UF0MODC	R/W		√		00H
00400078H	UF0 mode status register	UF0MODS	R		√		00H

(2/2)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400080H	UF0 active interface number register	UF0AIFN	R/W		√		00H
00400082H	UF0 active alternative setting register	UF0AAS	R/W		√		00H
00400084H	UF0 alternative setting status register	UF0ASS	R		√		00H
00400086H	UF0 endpoint 1 interface mapping register	UF0E1IM	R/W		√		00H
00400088H	UF0 endpoint 2 interface mapping register	UF0E2IM	R/W		√		00H
0040008AH	UF0 endpoint 3 interface mapping register	UF0E3IM	R/W		√		00H
0040008CH	UF0 endpoint 4 interface mapping register	UF0E4IM	R/W		√		00H
00400092H	UF0 endpoint 7 interface mapping register	UF0E7IM	R/W		√		00H

(2) EPC data hold register

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400100 H	UF0 EP0 read register	UF0E0R	R		√		Undefined
00400102H	UF0 EP0 length register	UF0E0L	R		√		00H
00400104H	UF0 EP0 setup register	UF0E0ST	R		√		00H
00400106H	UF0 EP0 write register	UF0E0W	W		√		Undefined
00400108H	UF0 bulk-out 1 register	UF0BO1	R		√		Undefined
0040010AH	UF0 bulk-out 1 length register	UF0BO1L	R		√		00H
0040010CH	UF0 bulk-out 2 register	UF0BO2	R		√		Undefined
0040010EH	UF0 bulk-out 2 length register	UF0BO2L	R		√		00H
00400110H	UF0 bulk-in 1 register	UF0BI1	W		√		Undefined
00400112H	UF0 bulk-in 2 register	UF0BI2	W		√		Undefined
00400114H	UF0 interrupt 1 register	UF0INT1	W		√		Undefined

(3) EPC request data register

(1/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400144H	UF0 device status register L	UF0DSTL	R/W		√		00H
0040014CH	UF0 EP0 status register L	UF0E0SL	R/W		√		00H
00400150H	UF0 EP1 status register L	UF0E1SL	R/W		√		00H
00400154H	UF0 EP2 status register L	UF0E2SL	R/W		√		00H
00400158H	UF0 EP3 status register L	UF0E3SL	R/W		√		00H
0040015CH	UF0 EP4 status register L	UF0E4SL	R/W		√		00H
00400168H	UF0 EP7 status register L	UF0E7SL	R/W		√		00H
00400180H	UF0 address register	UF0ADRS	R		√		00H
00400182H	UF0 configuration register	UF0CNF	R		√		00H
00400184H	UF0 interface 0 register	UF0IF0	R		√		00H
00400186H	UF0 interface 1 register	UF0IF1	R		√		00H
00400188H	UF0 interface 2 register	UF0IF2	R		√		00H
0040018AH	UF0 interface 3 register	UF0IF3	R		√		00H
0040018CH	UF0 interface 4 register	UF0IF4	R		√		00H
004001A0H	UF0 descriptor length register	UF0DSCL	R/W		√		00H
004001A2H	UF0 device descriptor register 0	UF0DD0	R/W		√		Undefined
004001A4H	UF0 device descriptor register 1	UF0DD1	R/W		√		Undefined
004001A6H	UF0 device descriptor register 2	UF0DD2	R/W		√		Undefined
004001A8H	UF0 device descriptor register 3	UF0DD3	R/W		√		Undefined
004001AAH	UF0 device descriptor register 4	UF0DD4	R/W		√		Undefined
004001ACH	UF0 device descriptor register 5	UF0DD5	R/W		√		Undefined
004001AEH	UF0 device descriptor register 6	UF0DD6	R/W		√		Undefined
004001B0H	UF0 device descriptor register 7	UF0DD7	R/W		√		Undefined
004001B2H	UF0 device descriptor register 8	UF0DD8	R/W		√		Undefined
004001B4H	UF0 device descriptor register 9	UF0DD9	R/W		√		Undefined
004001B6H	UF0 device descriptor register 10	UF0DD10	R/W		√		Undefined
004001B8H	UF0 device descriptor register 11	UF0DD11	R/W		√		Undefined
004001BAH	UF0 device descriptor register 12	UF0DD12	R/W		√		Undefined
004001BCH	UF0 device descriptor register 13	UF0DD13	R/W		√		Undefined
004001BEH	UF0 device descriptor register 14	UF0DD14	R/W		√		Undefined
004001C0H	UF0 device descriptor register 15	UF0DD15	R/W		√		Undefined
004001C2H	UF0 device descriptor register 16	UF0DD16	R/W		√		Undefined
004001C4H	UF0 device descriptor register 17	UF0DD17	R/W		√		Undefined
004001C6H	UF0 configuration/interface/endpoint descriptor register 0	UF0CIE0	R/W		√		Undefined
004001C8H	UF0 configuration/interface/endpoint descriptor register 1	UF0CIE1	R/W		√		Undefined
004001CAH	UF0 configuration/interface/endpoint descriptor register 2	UF0CIE2	R/W		√		Undefined
004001CCH	UF0 configuration/interface/endpoint descriptor register 3	UF0CIE3	R/W		√		Undefined

(2/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
004001CEH	UF0 configuration/interface/endpoint descriptor register 4	UF0CIE4	R/W		√		Undefined
004001D0H	UF0 configuration/interface/endpoint descriptor register 5	UF0CIE5	R/W		√		Undefined
004001D2H	UF0 configuration/interface/endpoint descriptor register 6	UF0CIE6	R/W		√		Undefined
004001D4H	UF0 configuration/interface/endpoint descriptor register 7	UF0CIE7	R/W		√		Undefined
004001D6H	UF0 configuration/interface/endpoint descriptor register 8	UF0CIE8	R/W		√		Undefined
004001D8H	UF0 configuration/interface/endpoint descriptor register 9	UF0CIE9	R/W		√		Undefined
004001DAH	UF0 configuration/interface/endpoint descriptor register 10	UF0CIE10	R/W		√		Undefined
004001DCH	UF0 configuration/interface/endpoint descriptor register 11	UF0CIE11	R/W		√		Undefined
004001DEH	UF0 configuration/interface/endpoint descriptor register 12	UF0CIE12	R/W		√		Undefined
004001E0H	UF0 configuration/interface/endpoint descriptor register 13	UF0CIE13	R/W		√		Undefined
004001E2H	UF0 configuration/interface/endpoint descriptor register 14	UF0CIE14	R/W		√		Undefined
004001E4H	UF0 configuration/interface/endpoint descriptor register 15	UF0CIE15	R/W		√		Undefined
004001E6H	UF0 configuration/interface/endpoint descriptor register 16	UF0CIE16	R/W		√		Undefined
004001E8H	UF0 configuration/interface/endpoint descriptor register 17	UF0CIE17	R/W		√		Undefined
004001EAH	UF0 configuration/interface/endpoint descriptor register 18	UF0CIE18	R/W		√		Undefined
004001ECH	UF0 configuration/interface/endpoint descriptor register 19	UF0CIE19	R/W		√		Undefined
004001EEH	UF0 configuration/interface/endpoint descriptor register 20	UF0CIE20	R/W		√		Undefined
004001F0H	UF0 configuration/interface/endpoint descriptor register 21	UF0CIE21	R/W		√		Undefined
004001F2H	UF0 configuration/interface/endpoint descriptor register 22	UF0CIE22	R/W		√		Undefined
004001F4H	UF0 configuration/interface/endpoint descriptor register 23	UF0CIE23	R/W		√		Undefined
004001F6H	UF0 configuration/interface/endpoint descriptor register 24	UF0CIE24	R/W		√		Undefined
004001F8H	UF0 configuration/interface/endpoint descriptor register 25	UF0CIE25	R/W		√		Undefined

(3/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
004001FAH	UF0 configuration/interface/endpoint descriptor register 26	UF0CIE26	R/W		√		Undefined
004001FCH	UF0 configuration/interface/endpoint descriptor register 27	UF0CIE27	R/W		√		Undefined
004001FEH	UF0 configuration/interface/endpoint descriptor register 28	UF0CIE28	R/W		√		Undefined
00400200H	UF0 configuration/interface/endpoint descriptor register 29	UF0CIE29	R/W		√		Undefined
00400202H	UF0 configuration/interface/endpoint descriptor register 30	UF0CIE30	R/W		√		Undefined
00400204H	UF0 configuration/interface/endpoint descriptor register 31	UF0CIE31	R/W		√		Undefined
00400206H	UF0 configuration/interface/endpoint descriptor register 32	UF0CIE32	R/W		√		Undefined
00400208H	UF0 configuration/interface/endpoint descriptor register 33	UF0CIE33	R/W		√		Undefined
0040020AH	UF0 configuration/interface/endpoint descriptor register 34	UF0CIE34	R/W		√		Undefined
0040020CH	UF0 configuration/interface/endpoint descriptor register 35	UF0CIE35	R/W		√		Undefined
0040020EH	UF0 configuration/interface/endpoint descriptor register 36	UF0CIE36	R/W		√		Undefined
00400210H	UF0 configuration/interface/endpoint descriptor register 37	UF0CIE37	R/W		√		Undefined
00400212H	UF0 configuration/interface/endpoint descriptor register 38	UF0CIE38	R/W		√		Undefined
00400214H	UF0 configuration/interface/endpoint descriptor register 39	UF0CIE39	R/W		√		Undefined
00400216H	UF0 configuration/interface/endpoint descriptor register 40	UF0CIE40	R/W		√		Undefined
00400218H	UF0 configuration/interface/endpoint descriptor register 41	UF0CIE41	R/W		√		Undefined
0040021AH	UF0 configuration/interface/endpoint descriptor register 42	UF0CIE42	R/W		√		Undefined
0040021CH	UF0 configuration/interface/endpoint descriptor register 43	UF0CIE43	R/W		√		Undefined
0040021EH	UF0 configuration/interface/endpoint descriptor register 44	UF0CIE44	R/W		√		Undefined
00400220H	UF0 configuration/interface/endpoint descriptor register 45	UF0CIE45	R/W		√		Undefined
00400222H	UF0 configuration/interface/endpoint descriptor register 46	UF0CIE46	R/W		√		Undefined
00400224H	UF0 configuration/interface/endpoint descriptor register 47	UF0CIE47	R/W		√		Undefined

(4/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400226H	UF0 configuration/interface/endpoint descriptor register 48	UF0CIE48	R/W		√		Undefined
00400228H	UF0 configuration/interface/endpoint descriptor register 49	UF0CIE49	R/W		√		Undefined
0040022AH	UF0 configuration/interface/endpoint descriptor register 50	UF0CIE50	R/W		√		Undefined
0040022CH	UF0 configuration/interface/endpoint descriptor register 51	UF0CIE51	R/W		√		Undefined
0040022EH	UF0 configuration/interface/endpoint descriptor register 52	UF0CIE52	R/W		√		Undefined
00400230H	UF0 configuration/interface/endpoint descriptor register 53	UF0CIE53	R/W		√		Undefined
00400232H	UF0 configuration/interface/endpoint descriptor register 54	UF0CIE54	R/W		√		Undefined
00400234H	UF0 configuration/interface/endpoint descriptor register 55	UF0CIE55	R/W		√		Undefined
00400236H	UF0 configuration/interface/endpoint descriptor register 56	UF0CIE56	R/W		√		Undefined
00400238H	UF0 configuration/interface/endpoint descriptor register 57	UF0CIE57	R/W		√		Undefined
0040023AH	UF0 configuration/interface/endpoint descriptor register 58	UF0CIE58	R/W		√		Undefined
0040023CH	UF0 configuration/interface/endpoint descriptor register 59	UF0CIE59	R/W		√		Undefined
0040023EH	UF0 configuration/interface/endpoint descriptor register 60	UF0CIE60	R/W		√		Undefined
00400240H	UF0 configuration/interface/endpoint descriptor register 61	UF0CIE61	R/W		√		Undefined
00400242H	UF0 configuration/interface/endpoint descriptor register 62	UF0CIE62	R/W		√		Undefined
00400244H	UF0 configuration/interface/endpoint descriptor register 63	UF0CIE63	R/W		√		Undefined
00400246H	UF0 configuration/interface/endpoint descriptor register 64	UF0CIE64	R/W		√		Undefined
00400248H	UF0 configuration/interface/endpoint descriptor register 65	UF0CIE65	R/W		√		Undefined
0040024AH	UF0 configuration/interface/endpoint descriptor register 66	UF0CIE66	R/W		√		Undefined
0040024CH	UF0 configuration/interface/endpoint descriptor register 67	UF0CIE67	R/W		√		Undefined
0040024EH	UF0 configuration/interface/endpoint descriptor register 68	UF0CIE68	R/W		√		Undefined
00400250H	UF0 configuration/interface/endpoint descriptor register 69	UF0CIE69	R/W		√		Undefined

(5/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400252H	UF0 configuration/interface/endpoint descriptor register 70	UF0CIE70	R/W		√		Undefined
00400254H	UF0 configuration/interface/endpoint descriptor register 71	UF0CIE71	R/W		√		Undefined
00400256H	UF0 configuration/interface/endpoint descriptor register 72	UF0CIE72	R/W		√		Undefined
00400258H	UF0 configuration/interface/endpoint descriptor register 73	UF0CIE73	R/W		√		Undefined
0040025AH	UF0 configuration/interface/endpoint descriptor register 74	UF0CIE74	R/W		√		Undefined
0040025CH	UF0 configuration/interface/endpoint descriptor register 75	UF0CIE75	R/W		√		Undefined
0040025EH	UF0 configuration/interface/endpoint descriptor register 76	UF0CIE76	R/W		√		Undefined
00400260H	UF0 configuration/interface/endpoint descriptor register 77	UF0CIE77	R/W		√		Undefined
00400262H	UF0 configuration/interface/endpoint descriptor register 78	UF0CIE78	R/W		√		Undefined
00400264H	UF0 configuration/interface/endpoint descriptor register 79	UF0CIE79	R/W		√		Undefined
00400266H	UF0 configuration/interface/endpoint descriptor register 80	UF0CIE80	R/W		√		Undefined
00400268H	UF0 configuration/interface/endpoint descriptor register 81	UF0CIE81	R/W		√		Undefined
0040026AH	UF0 configuration/interface/endpoint descriptor register 82	UF0CIE82	R/W		√		Undefined
0040026CH	UF0 configuration/interface/endpoint descriptor register 83	UF0CIE83	R/W		√		Undefined
0040026EH	UF0 configuration/interface/endpoint descriptor register 84	UF0CIE84	R/W		√		Undefined
00400270H	UF0 configuration/interface/endpoint descriptor register 85	UF0CIE85	R/W		√		Undefined
00400272H	UF0 configuration/interface/endpoint descriptor register 86	UF0CIE86	R/W		√		Undefined
00400274H	UF0 configuration/interface/endpoint descriptor register 87	UF0CIE87	R/W		√		Undefined
00400276H	UF0 configuration/interface/endpoint descriptor register 88	UF0CIE88	R/W		√		Undefined
00400278H	UF0 configuration/interface/endpoint descriptor register 89	UF0CIE89	R/W		√		Undefined
0040027AH	UF0 configuration/interface/endpoint descriptor register 90	UF0CIE90	R/W		√		Undefined
0040027CH	UF0 configuration/interface/endpoint descriptor register 91	UF0CIE91	R/W		√		Undefined

(6/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
0040027EH	UF0 configuration/interface/endpoint descriptor register 92	UF0CIE92	R/W		√		Undefined
00400280H	UF0 configuration/interface/endpoint descriptor register 93	UF0CIE93	R/W		√		Undefined
00400282H	UF0 configuration/interface/endpoint descriptor register 94	UF0CIE94	R/W		√		Undefined
00400284H	UF0 configuration/interface/endpoint descriptor register 95	UF0CIE95	R/W		√		Undefined
00400286H	UF0 configuration/interface/endpoint descriptor register 96	UF0CIE96	R/W		√		Undefined
00400288H	UF0 configuration/interface/endpoint descriptor register 97	UF0CIE97	R/W		√		Undefined
0040028AH	UF0 configuration/interface/endpoint descriptor register 98	UF0CIE98	R/W		√		Undefined
0040028CH	UF0 configuration/interface/endpoint descriptor register 99	UF0CIE99	R/W		√		Undefined
0040028EH	UF0 configuration/interface/endpoint descriptor register 100	UF0CIE100	R/W		√		Undefined
00400290H	UF0 configuration/interface/endpoint descriptor register 101	UF0CIE101	R/W		√		Undefined
00400292H	UF0 configuration/interface/endpoint descriptor register 102	UF0CIE102	R/W		√		Undefined
00400294H	UF0 configuration/interface/endpoint descriptor register 103	UF0CIE103	R/W		√		Undefined
00400296H	UF0 configuration/interface/endpoint descriptor register 104	UF0CIE104	R/W		√		Undefined
00400298H	UF0 configuration/interface/endpoint descriptor register 105	UF0CIE105	R/W		√		Undefined
0040029AH	UF0 configuration/interface/endpoint descriptor register 106	UF0CIE106	R/W		√		Undefined
0040029CH	UF0 configuration/interface/endpoint descriptor register 107	UF0CIE107	R/W		√		Undefined
0040029EH	UF0 configuration/interface/endpoint descriptor register 108	UF0CIE108	R/W		√		Undefined
004002A0H	UF0 configuration/interface/endpoint descriptor register 109	UF0CIE109	R/W		√		Undefined
004002A2H	UF0 configuration/interface/endpoint descriptor register 110	UF0CIE110	R/W		√		Undefined
004002A4H	UF0 configuration/interface/endpoint descriptor register 111	UF0CIE111	R/W		√		Undefined
004002A6H	UF0 configuration/interface/endpoint descriptor register 112	UF0CIE112	R/W		√		Undefined
004002A8H	UF0 configuration/interface/endpoint descriptor register 113	UF0CIE113	R/W		√		Undefined

(7/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
004002AAH	UF0 configuration/interface/endpoint descriptor register 114	UF0CIE114	R/W		√		Undefined
004002ACH	UF0 configuration/interface/endpoint descriptor register 115	UF0CIE115	R/W		√		Undefined
004002AEH	UF0 configuration/interface/endpoint descriptor register 116	UF0CIE116	R/W		√		Undefined
004002B0H	UF0 configuration/interface/endpoint descriptor register 117	UF0CIE117	R/W		√		Undefined
004002B2H	UF0 configuration/interface/endpoint descriptor register 118	UF0CIE118	R/W		√		Undefined
004002B4H	UF0 configuration/interface/endpoint descriptor register 119	UF0CIE119	R/W		√		Undefined
004002B6H	UF0 configuration/interface/endpoint descriptor register 120	UF0CIE120	R/W		√		Undefined
004002B8H	UF0 configuration/interface/endpoint descriptor register 121	UF0CIE121	R/W		√		Undefined
004002BAH	UF0 configuration/interface/endpoint descriptor register 122	UF0CIE122	R/W		√		Undefined
004002BCH	UF0 configuration/interface/endpoint descriptor register 123	UF0CIE123	R/W		√		Undefined
004002BEH	UF0 configuration/interface/endpoint descriptor register 124	UF0CIE124	R/W		√		Undefined
004002C0H	UF0 configuration/interface/endpoint descriptor register 125	UF0CIE125	R/W		√		Undefined
004002C2H	UF0 configuration/interface/endpoint descriptor register 126	UF0CIE126	R/W		√		Undefined
004002C4H	UF0 configuration/interface/endpoint descriptor register 127	UF0CIE127	R/W		√		Undefined
004002C6H	UF0 configuration/interface/endpoint descriptor register 128	UF0CIE128	R/W		√		Undefined
004002C8H	UF0 configuration/interface/endpoint descriptor register 129	UF0CIE129	R/W		√		Undefined
004002CAH	UF0 configuration/interface/endpoint descriptor register 130	UF0CIE130	R/W		√		Undefined
004002CCH	UF0 configuration/interface/endpoint descriptor register 131	UF0CIE131	R/W		√		Undefined
004002CEH	UF0 configuration/interface/endpoint descriptor register 132	UF0CIE132	R/W		√		Undefined
004002D0H	UF0 configuration/interface/endpoint descriptor register 133	UF0CIE133	R/W		√		Undefined
004002D2H	UF0 configuration/interface/endpoint descriptor register 134	UF0CIE134	R/W		√		Undefined
004002D4H	UF0 configuration/interface/endpoint descriptor register 135	UF0CIE135	R/W		√		Undefined

(8/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
004002D6H	UF0 configuration/interface/endpoint descriptor register 136	UF0CIE136	R/W		√		Undefined
004002D8H	UF0 configuration/interface/endpoint descriptor register 137	UF0CIE137	R/W		√		Undefined
004002DAH	UF0 configuration/interface/endpoint descriptor register 138	UF0CIE138	R/W		√		Undefined
004002DCH	UF0 configuration/interface/endpoint descriptor register 139	UF0CIE139	R/W		√		Undefined
004002DEH	UF0 configuration/interface/endpoint descriptor register 140	UF0CIE140	R/W		√		Undefined
004002E0H	UF0 configuration/interface/endpoint descriptor register 141	UF0CIE141	R/W		√		Undefined
004002E2H	UF0 configuration/interface/endpoint descriptor register 142	UF0CIE142	R/W		√		Undefined
004002E4H	UF0 configuration/interface/endpoint descriptor register 143	UF0CIE143	R/W		√		Undefined
004002E6H	UF0 configuration/interface/endpoint descriptor register 144	UF0CIE144	R/W		√		Undefined
004002E8H	UF0 configuration/interface/endpoint descriptor register 145	UF0CIE145	R/W		√		Undefined
004002EAH	UF0 configuration/interface/endpoint descriptor register 146	UF0CIE146	R/W		√		Undefined
004002ECH	UF0 configuration/interface/endpoint descriptor register 147	UF0CIE147	R/W		√		Undefined
004002EEH	UF0 configuration/interface/endpoint descriptor register 148	UF0CIE148	R/W		√		Undefined
004002F0H	UF0 configuration/interface/endpoint descriptor register 149	UF0CIE149	R/W		√		Undefined
004002F2H	UF0 configuration/interface/endpoint descriptor register 150	UF0CIE150	R/W		√		Undefined
004002F4H	UF0 configuration/interface/endpoint descriptor register 151	UF0CIE151	R/W		√		Undefined
004002F6H	UF0 configuration/interface/endpoint descriptor register 152	UF0CIE152	R/W		√		Undefined
004002F8H	UF0 configuration/interface/endpoint descriptor register 153	UF0CIE153	R/W		√		Undefined
004002FAH	UF0 configuration/interface/endpoint descriptor register 154	UF0CIE154	R/W		√		Undefined
004002FCH	UF0 configuration/interface/endpoint descriptor register 155	UF0CIE155	R/W		√		Undefined
004002FEH	UF0 configuration/interface/endpoint descriptor register 156	UF0CIE156	R/W		√		Undefined
00400300H	UF0 configuration/interface/endpoint descriptor register 157	UF0CIE157	R/W		√		Undefined

(9/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400302H	UF0 configuration/interface/endpoint descriptor register 158	UF0CIE158	R/W		√		Undefined
00400304H	UF0 configuration/interface/endpoint descriptor register 159	UF0CIE159	R/W		√		Undefined
00400306H	UF0 configuration/interface/endpoint descriptor register 160	UF0CIE160	R/W		√		Undefined
00400308H	UF0 configuration/interface/endpoint descriptor register 161	UF0CIE161	R/W		√		Undefined
0040030AH	UF0 configuration/interface/endpoint descriptor register 162	UF0CIE162	R/W		√		Undefined
0040030CH	UF0 configuration/interface/endpoint descriptor register 163	UF0CIE163	R/W		√		Undefined
0040030EH	UF0 configuration/interface/endpoint descriptor register 164	UF0CIE164	R/W		√		Undefined
00400310H	UF0 configuration/interface/endpoint descriptor register 165	UF0CIE165	R/W		√		Undefined
00400312H	UF0 configuration/interface/endpoint descriptor register 166	UF0CIE166	R/W		√		Undefined
00400314H	UF0 configuration/interface/endpoint descriptor register 167	UF0CIE167	R/W		√		Undefined
00400316H	UF0 configuration/interface/endpoint descriptor register 168	UF0CIE168	R/W		√		Undefined
00400318H	UF0 configuration/interface/endpoint descriptor register 169	UF0CIE169	R/W		√		Undefined
0040031AH	UF0 configuration/interface/endpoint descriptor register 170	UF0CIE170	R/W		√		Undefined
0040031CH	UF0 configuration/interface/endpoint descriptor register 171	UF0CIE171	R/W		√		Undefined
0040031EH	UF0 configuration/interface/endpoint descriptor register 172	UF0CIE172	R/W		√		Undefined
00400320H	UF0 configuration/interface/endpoint descriptor register 173	UF0CIE173	R/W		√		Undefined
00400322H	UF0 configuration/interface/endpoint descriptor register 174	UF0CIE174	R/W		√		Undefined
00400324H	UF0 configuration/interface/endpoint descriptor register 175	UF0CIE175	R/W		√		Undefined
00400326H	UF0 configuration/interface/endpoint descriptor register 176	UF0CIE176	R/W		√		Undefined
00400328H	UF0 configuration/interface/endpoint descriptor register 177	UF0CIE177	R/W		√		Undefined
0040032AH	UF0 configuration/interface/endpoint descriptor register 178	UF0CIE178	R/W		√		Undefined
0040032CH	UF0 configuration/interface/endpoint descriptor register 179	UF0CIE179	R/W		√		Undefined

(10/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
0040032EH	UF0 configuration/interface/endpoint descriptor register 180	UF0CIE180	R/W		√		Undefined
00400330H	UF0 configuration/interface/endpoint descriptor register 181	UF0CIE181	R/W		√		Undefined
00400332H	UF0 configuration/interface/endpoint descriptor register 182	UF0CIE182	R/W		√		Undefined
00400334H	UF0 configuration/interface/endpoint descriptor register 183	UF0CIE183	R/W		√		Undefined
00400336H	UF0 configuration/interface/endpoint descriptor register 184	UF0CIE184	R/W		√		Undefined
00400338H	UF0 configuration/interface/endpoint descriptor register 185	UF0CIE185	R/W		√		Undefined
0040033AH	UF0 configuration/interface/endpoint descriptor register 186	UF0CIE186	R/W		√		Undefined
0040033CH	UF0 configuration/interface/endpoint descriptor register 187	UF0CIE187	R/W		√		Undefined
0040033EH	UF0 configuration/interface/endpoint descriptor register 188	UF0CIE188	R/W		√		Undefined
00400340H	UF0 configuration/interface/endpoint descriptor register 189	UF0CIE189	R/W		√		Undefined
00400342H	UF0 configuration/interface/endpoint descriptor register 190	UF0CIE190	R/W		√		Undefined
00400344H	UF0 configuration/interface/endpoint descriptor register 191	UF0CIE191	R/W		√		Undefined
00400346H	UF0 configuration/interface/endpoint descriptor register 192	UF0CIE192	R/W		√		Undefined
00400348H	UF0 configuration/interface/endpoint descriptor register 193	UF0CIE193	R/W		√		Undefined
0040034AH	UF0 configuration/interface/endpoint descriptor register 194	UF0CIE194	R/W		√		Undefined
0040034CH	UF0 configuration/interface/endpoint descriptor register 195	UF0CIE195	R/W		√		Undefined
0040034EH	UF0 configuration/interface/endpoint descriptor register 196	UF0CIE196	R/W		√		Undefined
00400350H	UF0 configuration/interface/endpoint descriptor register 197	UF0CIE197	R/W		√		Undefined
00400352H	UF0 configuration/interface/endpoint descriptor register 198	UF0CIE198	R/W		√		Undefined
00400354H	UF0 configuration/interface/endpoint descriptor register 199	UF0CIE199	R/W		√		Undefined
00400356H	UF0 configuration/interface/endpoint descriptor register 200	UF0CIE200	R/W		√		Undefined
00400358H	UF0 configuration/interface/endpoint descriptor register 201	UF0CIE201	R/W		√		Undefined

(11/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
0040035AH	UF0 configuration/interface/endpoint descriptor register 202	UF0CIE202	R/W		√		Undefined
0040035CH	UF0 configuration/interface/endpoint descriptor register 203	UF0CIE203	R/W		√		Undefined
0040035EH	UF0 configuration/interface/endpoint descriptor register 204	UF0CIE204	R/W		√		Undefined
00400360H	UF0 configuration/interface/endpoint descriptor register 205	UF0CIE205	R/W		√		Undefined
00400362H	UF0 configuration/interface/endpoint descriptor register 206	UF0CIE206	R/W		√		Undefined
00400364H	UF0 configuration/interface/endpoint descriptor register 207	UF0CIE207	R/W		√		Undefined
00400366H	UF0 configuration/interface/endpoint descriptor register 208	UF0CIE208	R/W		√		Undefined
00400368H	UF0 configuration/interface/endpoint descriptor register 209	UF0CIE209	R/W		√		Undefined
0040036AH	UF0 configuration/interface/endpoint descriptor register 210	UF0CIE210	R/W		√		Undefined
0040036CH	UF0 configuration/interface/endpoint descriptor register 211	UF0CIE211	R/W		√		Undefined
0040036EH	UF0 configuration/interface/endpoint descriptor register 212	UF0CIE212	R/W		√		Undefined
00400370H	UF0 configuration/interface/endpoint descriptor register 213	UF0CIE213	R/W		√		Undefined
00400372H	UF0 configuration/interface/endpoint descriptor register 214	UF0CIE214	R/W		√		Undefined
00400374H	UF0 configuration/interface/endpoint descriptor register 215	UF0CIE215	R/W		√		Undefined
00400376H	UF0 configuration/interface/endpoint descriptor register 216	UF0CIE216	R/W		√		Undefined
00400378H	UF0 configuration/interface/endpoint descriptor register 217	UF0CIE217	R/W		√		Undefined
0040037AH	UF0 configuration/interface/endpoint descriptor register 218	UF0CIE218	R/W		√		Undefined
0040037CH	UF0 configuration/interface/endpoint descriptor register 219	UF0CIE219	R/W		√		Undefined
0040037EH	UF0 configuration/interface/endpoint descriptor register 220	UF0CIE220	R/W		√		Undefined
00400380H	UF0 configuration/interface/endpoint descriptor register 221	UF0CIE221	R/W		√		Undefined
00400382H	UF0 configuration/interface/endpoint descriptor register 222	UF0CIE222	R/W		√		Undefined
00400384H	UF0 configuration/interface/endpoint descriptor register 223	UF0CIE223	R/W		√		Undefined

(12/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400386H	UF0 configuration/interface/endpoint descriptor register 224	UF0CIE224	R/W		√		Undefined
00400388H	UF0 configuration/interface/endpoint descriptor register 225	UF0CIE225	R/W		√		Undefined
0040038AH	UF0 configuration/interface/endpoint descriptor register 226	UF0CIE226	R/W		√		Undefined
0040038CH	UF0 configuration/interface/endpoint descriptor register 227	UF0CIE227	R/W		√		Undefined
0040038EH	UF0 configuration/interface/endpoint descriptor register 228	UF0CIE228	R/W		√		Undefined
00400390H	UF0 configuration/interface/endpoint descriptor register 229	UF0CIE229	R/W		√		Undefined
00400392H	UF0 configuration/interface/endpoint descriptor register 230	UF0CIE230	R/W		√		Undefined
00400394H	UF0 configuration/interface/endpoint descriptor register 231	UF0CIE231	R/W		√		Undefined
00400396H	UF0 configuration/interface/endpoint descriptor register 232	UF0CIE232	R/W		√		Undefined
00400398H	UF0 configuration/interface/endpoint descriptor register 233	UF0CIE233	R/W		√		Undefined
0040039AH	UF0 configuration/interface/endpoint descriptor register 234	UF0CIE234	R/W		√		Undefined
0040039CH	UF0 configuration/interface/endpoint descriptor register 235	UF0CIE235	R/W		√		Undefined
0040039EH	UF0 configuration/interface/endpoint descriptor register 236	UF0CIE236	R/W		√		Undefined
004003A0H	UF0 configuration/interface/endpoint descriptor register 237	UF0CIE237	R/W		√		Undefined
004003A2H	UF0 configuration/interface/endpoint descriptor register 238	UF0CIE238	R/W		√		Undefined
004003A4H	UF0 configuration/interface/endpoint descriptor register 239	UF0CIE239	R/W		√		Undefined
004003A6H	UF0 configuration/interface/endpoint descriptor register 240	UF0CIE240	R/W		√		Undefined
004003A8H	UF0 configuration/interface/endpoint descriptor register 241	UF0CIE241	R/W		√		Undefined
004003AAH	UF0 configuration/interface/endpoint descriptor register 242	UF0CIE242	R/W		√		Undefined
004003ACH	UF0 configuration/interface/endpoint descriptor register 243	UF0CIE243	R/W		√		Undefined
004003AEH	UF0 configuration/interface/endpoint descriptor register 244	UF0CIE244	R/W		√		Undefined
004003B0H	UF0 configuration/interface/endpoint descriptor register 245	UF0CIE245	R/W		√		Undefined

(13/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
004003B2H	UF0 configuration/interface/endpoint descriptor register 246	UF0CIE246	R/W		√		Undefined
004003B4H	UF0 configuration/interface/endpoint descriptor register 247	UF0CIE247	R/W		√		Undefined
004003B6H	UF0 configuration/interface/endpoint descriptor register 248	UF0CIE248	R/W		√		Undefined
004003B8H	UF0 configuration/interface/endpoint descriptor register 249	UF0CIE249	R/W		√		Undefined
004003BAH	UF0 configuration/interface/endpoint descriptor register 250	UF0CIE250	R/W		√		Undefined
004003BCH	UF0 configuration/interface/endpoint descriptor register 251	UF0CIE251	R/W		√		Undefined
004003BEH	UF0 configuration/interface/endpoint descriptor register 252	UF0CIE252	R/W		√		Undefined
004003C0H	UF0 configuration/interface/endpoint descriptor register 253	UF0CIE253	R/W		√		Undefined
004003C2H	UF0 configuration/interface/endpoint descriptor register 254	UF0CIE254	R/W		√		Undefined
004003C4H	UF0 configuration/interface/endpoint descriptor register 255	UF0CIE255	R/W		√		Undefined

(4) Bridge register

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
00400400H	Bridge interrupt control register	BRGINTT	R/W			√	0000H
00400402H	Bridge interrupt enable register	BRGINTE	R/W			√	0000H
00400404H	EPC macro control register	EPCCLT	R/W			√	0000H
00400408H	CPU I/F bus control register	CPUBCTL	R/W			√	0000H

18.6.3 EPC control registers

(1) UF0 EP0NAK register (UF0E0N)

This register controls NAK of Endpoint0 (except an automatically executed request).

This register can be read or written in 8-bit units (however, bit 0 can only be read).

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the UF0FIC0 and UF0FIC1 registers from a read signal that accesses the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read, Endpoint2, and Endpoint4, a write access to the EP0NKR bit is ignored.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0N	0	0	0	0	0	0	EP0NKR	EP0NKW	00400000H	00H

Bit position	Bit name	Function
1	EP0NKR	This bit controls NAK to the OUT token to Endpoint0 (except an automatically executed request). It is automatically set to 1 by hardware when Endpoint0 has correctly received data. It is also cleared to 0 by hardware when the data of the UF0E0R register has been read by FW (counter value = 0). 1: Transmit NAK. 0: Do not transmit NAK (default value). Set this bit to 1 by FW when data should not be received from the USB bus for some reason even when USBF is ready for receiving data. In this case, USBF continues transmitting NAK until this bit is cleared to 0 by FW. This bit is also cleared to 0 as soon as the UF0E0R register has been cleared.
0	EP0NKW	This bit indicates how NAK to the IN token to Endpoint0 is controlled (except an automatically executed request). This bit is automatically cleared to 0 by hardware when the data of Endpoint0 is transmitted and the host correctly receives the transmitted data. The data of the UF0E0W register is retained until this bit is cleared. Therefore, it is not necessary to rewrite this bit even in the case of a retransmission request that is made if the host could not receive data correctly. To send a short packet, be sure to set the E0DED bit of the UF0DEND register to 1. This bit is automatically set to 1 when the FIFO is full. As soon as the E0DED bit of the UF0DEND register is set to 1, the EP0NKW bit is automatically set to 1 at the same time. 1: Do not transmit NAK. 0: Transmit NAK (default value). If control transfer enters the status stage while ACK cannot be correctly received in the data stage, this bit is cleared to 0 as soon as the UF0E0W register is cleared. This bit is also cleared to 0 when UF0E0W is cleared by FW.

Next, the procedure of a SETUP transaction that uses IN/OUT tokens is explained below.

(a) When IN token is used (except a request automatically executed by hardware)

FW should be used to clear the PROT bit of the UF0IS1 register to 0 after receiving the CPUDEC interrupt and before reading data from the UF0E0ST register. Next, perform processing in accordance with the request and, if it is necessary to return data by an IN token, write data to the UF0E0W register. Confirm that the PROT bit of the UF0IS1 register is 0 after writing has been completed, and set the E0DED bit of the UF0DEND register to 1. The hardware sends out data at the first IN token after the EP0NKW bit has been set to 1. If the PROT bit of the UF0IS1 register is 1, it indicates that a SETUP transaction has occurred again before completion of control transfer. In this case, clear the PROT bit of the UF0IS1 register to 0 by clearing the PROTC bit of the UF0IC1 register to 0, and then read data from the UF0E0ST register again. A request received later can be read.

(b) When OUT token is used (except a request automatically executed by hardware)

FW should be used to clear the PROT bit of the UF0IS1 register after receiving the CPUDEC interrupt and before reading data from the UF0E0ST register. Confirm that the PROT bit of the UF0IS1 register is 0 before reading data from the UF0E0R register. If the PROT bit is 1, it means that invalid data is retained. Clear the FIFO by FW (the EP0NKR bit is automatically cleared to 0). If the PROT bit of the UF0IS1 register is 0, read the data of the UF0E0L register and read as many data from the UF0E0R register as set. When reading data from the UF0E0R register has been completed (when the counter of the UF0E0R register has been cleared to 0), the hardware automatically clears the EP0NKR bit to 0.

(2) UF0 EP0NAKALL register (UF0E0NA)

This register controls NAK to all the requests of Endpoint0. It is also valid for automatically executed requests.

This register can be read or written in 8-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0NA	0	0	0	0	0	0	0	EP0NKA	00400002H	00H

Bit position	Bit name	Function
0	EP0NKA	This bit controls NAK to a transaction other than a SETUP transaction to Endpoint0 (including an automatically executed request). This bit is manipulated by FW. 1: Transmit NAK. 0: Do not transmit NAK (default value). This register is used to prevent a conflict between a write access by FW and a read access from SIE when the data used for an automatically executed request is to be changed. It postpones reflecting a write access on this bit from FW while an access from SIE is being made. Before rewriting the request data register from FW, confirm that this bit has been correctly set to 1. Setting this bit to 1 is reflected only in the following cases. • Immediately after USBF has been reset and a SETUP token has never been received • Immediately after reception of Bus Reset and a SETUP token has never been received • PID of a SETUP token has been detected • The stage has been changed to the status stage Clearing this bit to 0 is reflected immediately, except while an IN token is being received and a NAK response is being made. Setting the EP0NKA bit to 1 is reflected in the above four cases during Endpoint0 transfer, but it is reflected immediately after data has been written to the bit while Endpoint0 is transferring no data.

(3) UF0 EPNAK register (UF0EN)

This register controls NAK of endpoints other than Endpoint0.

This register can be read or written in 8-bit units (however, bits 5, 4, 1, and 0 can only be read).

The BKO2NK bit can be written only when the BKO2NKM bit of the UF0ENM register is 1 and the BKO1NK bit can be written only when the BKO1NKM bit of the UF0ENM register is 1.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1 to 4, 7) and the current setting of the interface.

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the UF0FIC0 and UF0FIC1 registers from a read signal that accesses the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read, Endpoint2, and Endpoint4, a write access to the BKO1NK and BKO2NK bits is ignored.

Be sure to clear bits 5 to 7 to "0". If it is set to 1, the operation is not guaranteed.

(1/3)

	7	6		4	3	2	1	0	Address	After reset
UF0EN	0	0	0	IT1NK	BKO2NK	BKO1NK	BKI2NK	BKI1NK	00400004H	00H

Bit position	Bit name	Function
4	IT1NK	This bit controls NAK to Endpoint7 (interrupt 1 transfer). It is automatically set to 1 and transmission is started when the UF0INT1 register has become full as a result of writing data to it. To send a short packet that does not make the FIFO full, set the IT1DEND bit of the UF0DEND register to 1. As soon as the IT1DEND bit has been set to 1, this bit is automatically set to 1. 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is also cleared to 0 when the UF0INT1 register has been cleared.

(2/3)

Bit position	Bit name	Function
3	BKO2NK	This bit controls NAK to Endpoint4 (bulk 2 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK (default value). This bit is set to 1 only when the FIFO connected to the SIE side of the UF0BO2 register (64-byte FIFO of bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied. • Data correctly received is stored in the FIFO connected to the SIE side. • The value of the FIFO counter connected to the CPU side is 0 (completion of reading). FW should be used to read data of the UF0BO2L register when it has received the BLKO2DT interrupt request and read as many data from the UF0BO2 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 by FW. In this case, USBF keeps transmitting NAK until the FW clears this bit to 0. This bit is also cleared to 0 as soon as the UF0BO2 register has been cleared.
2	BKO1NK	This bit controls NAK to Endpoint2 (bulk 1 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK (default value). This bit is set to 1 only when the FIFO connected to the SIE side of the UF0BO1 register (64-byte FIFO of bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied. • Data correctly received is stored in the FIFO connected to the SIE side. • The value of the FIFO counter connected to the CPU side is 0 (completion of reading). FW should be used to read data of the UF0BO1L register when it has received the BLKO1DT interrupt request and read as many data from the UF0BO1 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 by FW. In this case, USBF keeps transmitting NAK until the FW clears this bit to 0. This bit is also cleared to 0 as soon as the UF0BO1 register has been cleared.

(3/3)

Bit position	Bit name	Function
1	BKI2NK	This bit controls NAK to Endpoint3 (bulk 2 transfer (IN)). 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is cleared to 0 only when the FIFO connected to the SIE side of the UF0BI2 register (64-byte FIFO of bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed (the data of the UF0BI2 register is retained until transmission has been correctly completed). The bank is changed (toggle operation) when the following conditions are satisfied. • Data is correctly written to the FIFO connected to the CPU bus side (writing has been completed and the FIFO is full or the UF0DEND register is set). • The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO. However, if the FIFO on the CPU side becomes full as a result of writing data to it while the BKI2T bit of the UF0DEND register is cleared to 0, the toggle operation is not performed because the condition of the toggle operation is not satisfied until the BKI2DED bit of the UF0DEND register is set to 1. To send a short packet that does not make the FIFO on the CPU side full, set the BKI2DED bit to 1 after completing writing data. When the BKI2DED bit is set to 1, a toggle operation is performed and at the same time, this bit is automatically set to 1. This bit is also cleared to 0 as soon as the UF0BI2 register has been cleared.
0	BKI1NK	This bit controls NAK to Endpoint1 (bulk 1 transfer (IN)). 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is cleared to 0 only when the FIFO connected to the SIE side of the UF0BI1 register (64-byte FIFO of bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed (the data of the UF0BI1 register is retained until transmission has been correctly completed). The bank is changed (toggle operation) when the following conditions are satisfied. • Data is correctly written to the FIFO connected to the CPU bus side (writing has been completed and the FIFO is full or the UF0DEND register is set). • The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO. However, if the FIFO on the CPU side becomes full as a result of writing data to it while the BKI1T bit of the UF0DEND register is cleared to 0, the toggle operation is not performed because the condition of the toggle operation is not satisfied until the BKI1DED bit of the UF0DEND register is set to 1. To send a short packet that does not make the FIFO on the CPU side full, set the BKI1DED bit to 1 after completing writing data. When the BKI1DED bit is set to 1, a toggle operation is performed and at the same time, this bit is automatically set to 1. This bit is also cleared to 0 as soon as the UF0BI1 register has been cleared.

(4) UF0 EPNAK mask register (UF0ENM)

This register controls masking a write access to the UF0EN register.

This register can be read or written in 8-bit units.

Be sure to clear bits 0, 1, and 4 to 7 to “0”. If it is set to 1, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ENM	0	0	0	0	BKO2NKM	BKO1NKM	0	0	00400006H	00H

Bit position	Bit name	Function
3	BKO2NKM	This bit specifies whether a write access to bit 3 (BKO2NK) of the UF0EN register is masked or not. 1: Do not mask. 0: Mask (default value).
2	BKO1NKM	This bit specifies whether a write access to bit 2 (BKO1NK) of the UF0EN register is masked or not. 1: Do not mask. 0: Mask (default value).

(5) UF0 SNDSIE register (UF0SDS)

This register performs manipulation such as no handshake. It can directly manipulate the pins of SIE.

This register can be read or written in 8-bit units.

Be sure to clear bits 1, 2, and 4 to 7 to “0”. If it is set to 1, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0SDS	0	0	0	0	SNDSTL	0	0	RSUMIN	00400008H	00H

Bit position	Bit name	Function
3	SNDSTL	This bit makes Endpoint0 issue a STALL handshake. Setting this bit to 1 if a request for CPUDEC processing is not supported by the system results in a STALL handshake response. If an unsupported wValue is sent by the SET_CONFIGURATION or SET_INTERFACE request, the hardware sets this bit to 1. If a problem occurs in Endpoint0 due to overrun of an automatically executed request, this bit is also set to 1. However, the E0HALT bit of the UF0E0SL register is not set to 1. 1: Respond with STALL handshake. 0: Do not respond with STALL handshake (default value). This bit is cleared to 0 and the handshake response to the bus is other than STALL when the next SETUP token is received. To set the SNDSTL bit to 1 by FW, do not write data to the UF0E0W register. Depending on the timing of setting this bit, the STALL response is not made in time, and it may be made to the next transfer after a NAK response has been made. Setting this bit is valid only while an FW-executed request is under execution when this bit is set to 1. It is automatically cleared to 0 when the next SETUP token is received. **Remark** The SNDSTL bit is valid only for an FW-executed request.
0	RSUMIN	This bit outputs the Resume signal onto the USB bus. Writing this bit is invalid unless the RMWK bit of the UF0DSTL register is set to 1. 1: Generate the Resume signal. 0: Do not generate the Resume signal (default value). While this bit is set to 1, the Resume signal continues to be generated. Clear this bit to 0 by FW after a specific time has elapsed. Because the signal is internally sampled at the clock, the operation is guaranteed only while CLK is supplied. Care must be exercised when CLK of the system is stopped.

(6) UF0 CLR request register (UF0CLR)

This register indicates the target of the received CLEAR_FEATURE request.

This register is read-only, in 8-bit units.

This register is meaningful only when an interrupt request is generated. Each bit is set to 1 after completion of the status stage, and automatically cleared to 0 when this register is read.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1 to 4, 7) and the current setting of the interface.

Be sure to clear bit 7 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CLR	0	CLREP7	CLREP4	CLREP3	CLREP2	CLREP1	CLREP0	CLRDEV	0040000AH	00H

Bit position	Bit name	Function
6 to 1	CLREPn	These bits indicate that a CLEAR_FEATURE Endpoint n request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
0	CLRDEV	This bit indicates that a CLEAR_FEATURE Device request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)

Remark n = 0 to 4, 7

(7) UF0 SET request register (UF0SET)

This register indicates the target of the automatically processed SET_XXXX (except SET_INTERFACE) request.

This register is read-only, in 8-bit units.

This register is meaningful only when an interrupt request is generated. Each bit is set to 1 after completion of the status stage, and automatically cleared to 0 when this register is read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0SET	SETCON	0	0	0	0	SETEP	0	SETDEV	0040000CH	00H

Bit position	Bit name	Function
7	SETCON	This bit indicates that a SET_CONFIGURATION request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
2	SETEP	This bit indicates that a SET_FEATURE Endpoint n request (n = 0 to 4, 7) is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
0	SETDEV	This bit indicates that a SET_FEATURE Device request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)

(8) UF0 EP status 0 register (UF0EPS0)

This register indicates the USB bus status and the presence or absence of register data.

This register is read-only, in 8-bit units.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1 to 4, 7) and the current setting of the interface.

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate writing to the UF0FIC0 and UF0FIC1 registers from reading from the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

Be sure to clear bit 7 to "0".

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS0	0	IT1	BKOUT2	BKOUT1	BKIN2	BKIN1	EP0W	EP0R	0040000EH	00H

Bit position	Bit name	Function
6	IT1	This bit indicates that data is in the UF0INT1 register (FIFO). By setting the IT1DED bit of the UF0DEND register to 1, the status in which data is in the UF0INT1 register can be created even if data is not written to the register (Null data transmission). As soon as the IT1DED bit of the UF0DEND register is set to 1 even when the counter of the UF0INTn register is 0, this bit is set to 1 by hardware. It is cleared to 0 after correct transmission. 1: Data is in the register. 0: No data is in the register (default value).
5, 4	BKOUTn	These bits indicate that data is in the UF0BOn register (FIFO) connected to the CPU side. When the FIFO configuring the UF0BOn register is toggled, this bit is automatically set to 1 by hardware. It is automatically cleared to 0 by hardware when reading the UF0BOn register (FIFO) connected to the CPU side has been completed (counter value = 0). It is not set to 1 when Null data is received (toggling the FIFO does not take place either). 1: Data is in the register. 0: No data is in the register (default value).
3, 2	BKINn	These bits indicate that data is in the UF0BIn register (FIFO) connected to the CPU side. By setting the BKInDED bit of the UF0DEND register to 1, the status in which data is in the UF0BIn register can be created even if data is not written to the register (Null data transmission). As soon as the BKInDED bit of the UF0DEND register has been set to 1 while the counter of the UF0BIn register is 0, this bit is set to 1 by hardware. It is cleared to 0 when a toggle operation is performed. 1: Data is in the register. 0: No data is in the register (default value).

Remark n = 1, 2

(2/2)

Bit position	Bit name	Function
1	EP0W	This bit indicates that data is in the UF0E0W register (FIFO). By setting the E0DED bit of the UF0DEND register to 1, the status in which data is in the UF0E0W register can be created even if data is not written to the register (Null data transmission). As soon as the E0DED bit of the UF0DEND register is set to 1 even when the counter of the UF0E0W register is 0, this bit is set to 1 by hardware. It is cleared to 0 after correct transmission. 1: Data is in the register. 0: No data is in the register (default value).
0	EP0R	This bit indicates that data is in the UF0E0R register (FIFO). It is automatically cleared to 0 by hardware when reading the UF0E0R register (FIFO) has been completed (counter value = 0). It is not set to 1 if Null data is received. 1: Data is in the register. 0: No data is in the register (default value).

(9) UF0 EP status 1 register (UF0EPS1)

This register indicates the USB bus status and the presence or absence of register data.

This register is read-only, in 8-bit units.

Be sure to clear bits 0 to 6 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS1	RSUM	0	0	0	0	0	0	0	00400010H	00H

Bit position	Bit name	Function
7	RSUM	This bit indicates that the USB bus is in the Resume status. This bit is meaningful only when an interrupt request is generated. 1: Suspend status 0: Resume status (default value) Because sampling is internally performed with the clock, the operation is guaranteed only when CLK is supplied. Care must be exercised when CLK of the system is stopped. The INTUSBF1 signal of SIE operates even when CLK is stopped. It can therefore be supported by making the interrupt control register (UFIC1) valid or lowering the frequency of CLK to the USBF. This bit is automatically cleared to 0 when it is read.

(10) UF0 EP status 2 register (UF0EPS2)

This register indicates the USB bus status and the presence or absence of register data.

This register is read-only, in 8-bit units.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1 to 4, 7) and the current setting of the interface.

Be sure to clear bits 6 and 7 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS2	0	0	HALT7	HALT4	HALT3	HALT2	HALT1	HALT0	00400012H	00H

Bit position	Bit name	Function
5 to 0	HALTn	These bits indicate that Endpoint n is currently stalled. They are set to 1 when a stall condition, such as occurrence of an overrun and reception of an undefined request, is satisfied. These bits are automatically set to 1 by hardware. 1: Endpoint is stalled. 0: Endpoint is not stalled (default value). The SNDSTL bit is set to 1 as soon as the HALT0 bit has been set to 1 as a result of occurrence of an overrun or reception of an undefined request. If the next SETUP token is received in this status, the SNDSTL bit is cleared to 0 and, therefore, the HALT0 bit is also cleared to 0. If Endpoint0 is stalled by the SET_FEATURE Endpoint0 request, this bit is not cleared to 0 until the CLEAR_FEATURE Endpoint0 request is received or Halt Feature is cleared by FW. If the GET_STATUS Endpoint0, CLEAR_FEATURE Endpoint0, or SET_FEATURE Endpoint0 request is received, or if a request to be processed by FW is received due to the CPUDEC interrupt request, the HALT0 bit is masked and cleared to 0, until the next SETUP token is received. The HALTn bit is not cleared to 0 until Endpoint n receives the CLEAR_FEATURE Endpoint request, Halt Feature is cleared by the SET_INTERFACE or SET_CONFIGURATION request to the interface to which the endpoint is linked, or Halt Feature is cleared by FW. When the SET_INTERFACE or SET_CONFIGURATION request is correctly processed, the Halt Feature of all the target endpoints, except Endpoint0, is cleared after the request has been processed, even if the wValue is the same as the currently set value, and these bits are also cleared to 0. Halt Feature of Endpoint0 cannot be cleared if it is set because the STALL response is made in response to the SET_INTERFACE and SET_CONFIGURATION requests.

Remark n = 0 to 4, 7

(11) UF0 INT status 0 register (UF0IS0)

This register indicates the interrupt source. If the contents of this register are changed, the EPCINT0B becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSBF0) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC0 register.

Be sure to clear bits 3 to 5 to “0”.

Caution In the USBF, multiple interrupt sources, such as Bus Reset, and Resume, are ORed internally and are issued as a single interrupt request (INTUSBF0). Therefore, in the case of the occurrence of multiple interrupt sources, they are ORed and issued as an INTUSBF0 interrupt request.

For example, if a Bus Reset interrupt source and Resume interrupt source occur, the two sources are ORed and an INTUSBF0 interrupt request is issued.

Under these conditions, if the Bus Reset interrupt source is cleared to 0 (UF0IC0.BUSRSTC = 0), the V850E/IG4-H or V850E/IH4-H internal INTUSBF0 interrupt request may remain set to 1 since the Resume interrupt source will still remain. The new interrupt request flag (US0BIC.US0BIF), therefore, might not be set to 1.

In this case, after performing clear processing for each interrupt request with the INTUSBF0 interrupt servicing routine, confirm the flag status for the UF0IS0 and UF0IS1 registers again, and if there are any interrupt sources with flags set to 1, perform flag clearing (only the applicable bits need to be cleared (do not perform a batch clearing)).

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS0	BUSRST	RSUSPD	0	0	0	SETRQ	CLRRQ	EPHALT	00400020H	00H

Bit position	Bit name	Function
7	BUSRST	This bit indicates that Bus Reset has occurred. 1: Bus Reset has occurred (interrupt request is generated). 0: Not Bus Reset status (default value)
6	RSUSPD	This bit indicates that the Resume or Suspend status has occurred. Reference bit 7 of the UF0EPS1 register by FW. 1: Resume or Suspend status has occurred (interrupt request is generated). 0: Resume or Suspend status has not occurred (default value).
2	SETRQ	This bit indicates that the SET_XXXX request to be automatically processed has been received and automatically processed (XXXX = CONFIGURATION or FEATURE). 1: SET_XXXX request to be automatically processed has been received (interrupt request is generated). 0: SET_XXXX request to be automatically processed has not been received (default value). This bit is set to 1 after completion of the status stage. Reference the UF0SET register to identify what is the target of the request. This bit is not automatically cleared to 0 even if the UF0SET register is read by FW. The EPHALT bit is also set to 1 when the SET_FEATURE Endpoint request has been received.

(2/2)

Bit position	Bit name	Function
1	CLRRQ	This bit indicates that the CLEAR_FEATURE request has been received and automatically processed. 1: CLEAR_FEATURE request has been received (interrupt request is generated). 0: CLEAR_FEATURE request has not been received (default value). This bit is set to 1 after completion of the status stage. Reference the UF0CLR register to identify what is the target of the request. This bit is not automatically cleared to 0 even if the UF0CLR register is read by FW.
0	EPHALT	This bit indicates that an endpoint has stalled. 1: Endpoint has stalled (interrupt request is generated). 0: Endpoint has not stalled (default value). This bit is also set to 1 when an endpoint has stalled by setting FW. Identify the endpoint that has stalled, by referencing the UF0EPS2 register. This bit is not automatically cleared to 0 even when the CLEAR_FEATURE Endpoint, SET_INTERFACE, or SET_CONFIGURATION request is received. It is not automatically cleared to 0, either, if the next SETUP token is received in case of overrun of Endpoint0. **Caution Even if Halt Feature of Endpoint0 is set and this interrupt request is generated, bit 0 of the UF0EPS2 register is masked and cleared to 0 between when a SET_FEATURE Endpoint0, CLEAR_FEATURE Endpoint0, or GET_STATUS Endpoint0 request, or FW-processed request is received and when a SETUP token other than the above is received.**

(12) UF0 INT status 1 register (UF0IS1)

This register indicates the interrupt source. If the contents of this register are changed, the EPCINT0B becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSBF0) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC1 register. However, the SUCES and STG bits of the UF0IS1 register are automatically cleared to 0 when the next SETUP token has been received.

Caution In the USBF, multiple interrupt sources, such as Bus Reset, and Resume, are ORed internally and are issued as a single interrupt request (INTUSBF0). Therefore, in the case of the occurrence of multiple interrupt sources, they are ORed and issued as an INTUSBF0 interrupt request.

For example, if a Bus Reset interrupt source and Resume interrupt source occur, the two sources are ORed and an INTUSBF0 interrupt request is issued.

Under these conditions, if the Bus Reset interrupt source is cleared to 0 (UF0IC0.BUSRSTC = 0), the V850E/IG4-H or V850E/IH4-H internal INTUSBF0 interrupt request may remain set to 1 since the Resume interrupt source will still be remaining. The new interrupt request flag (US0BIC.US0BIF), therefore, might not be set to 1.

In this case, after performing clear processing for each interrupt request with the INTUSBF0 interrupt servicing routine, confirm the flag status for the UF0IS0 and UF0IS1 registers again, and if there are any interrupt sources with flags set to 1, perform flag clearing (only the applicable bits need to be cleared (do not perform a batch clearing)).

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS1	0	E0IN	E0INDT	E0ODT	SUCES	STG	PROT	CPU DEC	00400022H	00H

Bit position	Bit name	Function
6	E0IN	This bit indicates that an IN token for Endpoint0 has been received and that the hardware has automatically transmitted NAK. 1: IN token is received and NAK is transmitted (interrupt request is generated). 0: IN token is not received (default value).
5	E0INDT	This bit indicates that data has been correctly transmitted from the UF0E0W register. 1: Transmission from UF0E0W register is completed (interrupt request is generated). 0: Transmission from UF0E0W register is not completed (default value). Data is transmitted in synchronization with the IN token next to the one that set the EP0NKW bit of the UF0E0N register to 1. This bit is automatically set to 1 by hardware when the host correctly receives that data. It is also set to 1 even if the data is a Null packet. This bit is automatically cleared to 0 by hardware when the first write access is made to the UF0E0W register.

(2/2)

Bit position	Bit name	Function
4	E0ODT	This bit indicates that data has been correctly received in the UF0E0R register. 1: Data is in UF0E0R register (interrupt request is generated). 0: Data is not in UF0E0R register (default value). This bit is automatically set to 1 by hardware when data has been correctly received. At the same time, the EP0R bit of the UF0EPS0 register is also set to 1. If a Null packet has been received, this bit is not set to 1. It is automatically cleared to 0 by hardware when the FW reads the UF0E0R register and the value of the UF0E0L register becomes 0.
3	SUCES	This bit indicates that either an FW-processed or hardware-processed request has been received and that the status stage has been correctly completed. 1: Control transfer has been correctly processed (interrupt request is generated). 0: Control transfer has not been processed correctly (default value). This bit is set to 1 upon completion of the status stage. It is automatically cleared to 0 by hardware when the next SETUP token is received. This bit is also set to 1 when data with Data PID of 0 (Null data) is received in the status stage of control transfer.
2	STG	This bit is set to 1 when the stage of control transfer has changed to the status stage. It is valid for both FW-processed and hardware-processed requests. This bit is also set to 1 when the stage of control transfer (without data) has changed to the status stage. 1: Status stage (interrupt request is generated) 0: Not status stage (default value) This bit is automatically cleared to 0 by hardware when the next SETUP token is received. It is also set to 1 when the stage of control transfer has changed to the status stage while ACK cannot be correctly received in the data stage. In this case, the EP0NKW bit of the UF0E0N register is also cleared to 0 as soon as the UF0E0W register has been cleared, if the FW is processing control transfer (read).
1	PROT	This bit indicates that a SETUP token has been received. It is valid for both FW-processed and hardware-processed requests. 1: SETUP token is correctly received (interrupt request is generated). 0: SETUP token is not received (default value). This bit is set to 1 when data has been correctly received in the UF0E0ST register. Clear this bit to 0 by FW when the first read access is made to the UF0E0ST register. If it is not cleared to 0 by FW, reception of the next SETUP token cannot be correctly recognized. This bit is used to accurately recognize that a SETUP transaction has been executed again during control transfer. If the SETUP transaction is re-executed during control transfer and if a second request is executed by hardware, the CPUDEC bit is not set to 1, but the PROT bit can be used for recognition of the re-execution.
0	CPUDEC	This bit indicates that the UF0E0ST register has a request that is to be decoded by FW. 1: Data is in UF0E0ST register (interrupt request is generated). 0: Data is not in UF0E0ST register (default value). This bit is automatically cleared to 0 by hardware when all the data of the UF0E0ST register is read.

(13) UF0 INT status 2 register (UF0IS2)

This register indicates the interrupt source. If the contents of this register are changed, the EPCINT1B becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSBF0) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC2 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7) and the current setting of the interface.

Be sure to clear bits 1 to 3 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS2	BKI2IN	BKI2DT	BKI1IN	BKI1DT	0	0	0	IT1DT	00400024H	00H

Bit position	Bit name	Function
7, 5	BKInIN	These bits indicate that an IN token has been received in the UF0BIn register (Endpoint m) and that NAK has been returned. 1: IN token is received and NAK is transmitted (interrupt request is generated). 0: IN token is not received (default value).
6, 4	BKInDT	These bits indicate that the FIFO of the UF0BIn register (Endpoint m) has been toggled. This means that data can be written to Endpoint m. 1: FIFO has been toggled (interrupt request is generated). 0: FIFO has not been toggled (default value). The data written to Endpoint m is transmitted in synchronization with the IN token next to the one that set the BKInNK bit of the UF0EN register to 1. When the FIFO has been toggled and then data can be written from the CPU, this bit is automatically set to 1 by hardware. It is also set to 1 when the FIFO has been toggled, even if the data is a Null packet. This bit is automatically cleared to 0 by hardware when the first write access is made to the UF0BIn register.
0	IT1DT	This bit indicates that data has been correctly received from the UF0INT1 register (Endpoint 7). 1: Transmission is completed (interrupt request is generated). 0: Transmission is not completed (default value). Data is transmitted in synchronization with the IN token next to the one that set the IT1NK bit of the UF0EN register to 1. This bit is automatically set to 1 by hardware when the host has correctly received that data. It is automatically cleared to 0 by hardware when the first write access is made to the UF0INT1 register. This bit is also set to 1 even when the data is a Null packet.

Remark n = 1, 2
m = 1 where n = 1
m = 3 where n = 2

(14) UF0 INT status 3 register (UF0IS3)

This register indicates the interrupt source. If the contents of this register are changed, the EPCINT1B becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSBF0) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC3 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2, 4) and the current setting of the interface.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS3	BKO2FL	BKO2NL	BKO2 NAK	BKO2DT	BKO1FL	BKO1NL	BKO1 NAK	BKO1DT	00400026H	00H

Bit position	Bit name	Function
7, 3	BKOnFL	These bits indicate that data has been correctly received in the UF0BOn register (Endpoint m) and that both the FIFOs of the CPU and SIE hold the data. 1: Received data is in both the FIFOs of the UF0BOn register (interrupt request is generated). 0: Received data is not in the FIFO on the SIE side of the UF0BOn register (default value). If data is held in both the FIFOs of the CPU and SIE, these bits are automatically set to 1 by hardware. They are automatically cleared to 0 by hardware when the FIFO is toggled.
6, 2	BKOnNL	These bits indicate that a Null packet (packet with a length of 0) has been received in the UF0BOn register (Endpoint m). 1: Null packet is received (interrupt request is generated). 0: Null packet is not received (default value). These bits are set to 1 immediately after reception of a Null packet when the FIFO is empty. They are set to 1 when the FIFO on the CPU side has been completely read if data is in that FIFO.
5, 1	BKOnNAK	These bits indicate that an OUT token has been received to the UF0BOn register (Endpoint m) and that NAK has been returned. 1: OUT token is received and NAK is transmitted (interrupt request is generated). 0: OUT token is not received (default value).

Remark n = 1, 2
m = 2 where n = 1
m = 4 where n = 2

(2/2)

Bit position	Bit name	Function
4, 0	BKOnDT	These bits indicate that data has been correctly received in the UF0BOn register (Endpoint m). 1: Reception has been completed correctly (interrupt request is generated). 0: Reception has not been completed (default value). These bits are automatically set to 1 by hardware when data has been correctly received and the FIFO has been toggled. At the same time, the corresponding bits of the UF0EPS0 register are also set to 1. They are not set to 1 when the data is a Null packet. These bits are automatically cleared to 0 by hardware when the value of the UF0BOnL register becomes 0 as a result of reading the UF0BOn register by FW. These bits are automatically cleared to 0 when all the contents of the FIFO on the CPU side have been read. However, the interrupt request is not cleared if data is in the FIFO on the SIE side at this time, and the INTUSBF1 signal does not become inactive. The signal is kept active if data is successively received.

Remark n = 1, 2
m = 2 where n = 1
m = 4 where n = 2

(15) UF0 INT status 4 register (UF0IS4)

This register indicates the interrupt source. If the contents of this register are changed, the EPCINT2B becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSBF0) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC4 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1 to 4, 7) and the current setting of the interface.

Be sure to clear bits 0 to 4, 6, and 7 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS4	0	0	SETINT	0	0	0	0	0	00400028H	00H

Bit position	Bit name	Function
5	SETINT	This bit indicates that the SET_INTERFACE request has been received and automatically processed. 1: The request has been automatically processed (interrupt request is generated). 0: The request has not been automatically processed (default value). The current setting of this bit can be identified by reading the UF0ASS or UF0IFn register (n = 0 to 4).

(16) UF0 INT mask 0 register (UF0IM0)

This register controls masking of the interrupt sources indicated by the UF0IS0 register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

Be sure to clear bits 3 to 5 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM0	BUS RSTM	RSU SPDM	0	0	0	SET RQM	CLR RQM	EP HALTM	0040002EH	00H

Bit position	Bit name	Function
7	BUSRSTM	This bit masks the Bus Reset interrupt. 1: Mask 0: Do not mask (default value)
6	RSUSPDM	This bit masks the Resume/Suspend interrupt. 1: Mask 0: Do not mask (default value)
2	SETRQM	This bit masks the SET_RQ interrupt. 1: Mask 0: Do not mask (default value)
1	CLRRQM	This bit masks the CLR_RQ interrupt. 1: Mask 0: Do not mask (default value)
0	EPHALTM	This bit masks the EP_Halt interrupt. 1: Mask 0: Do not mask (default value)

(17) UF0 INT mask 1 register (UF0IM1)

This register controls masking of the interrupt sources indicated by the UF0IS1 register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM1	0	E0INM	E0 INDTM	E0 ODTM	SUCESM	STGM	PROTM	CPU DECM	00400030H	00H

Bit position	Bit name	Function
6	E0INM	This bit masks the EP0IN interrupt. 1: Mask 0: Do not mask (default value)
5	E0INDTM	This bit masks the EP0INDT interrupt. 1: Mask 0: Do not mask (default value)
4	E0ODTM	This bit masks the EP0OUTDT interrupt. 1: Mask 0: Do not mask (default value)
3	SUCESM	This bit masks the Success interrupt. 1: Mask 0: Do not mask (default value)
2	STGM	This bit masks the Stg interrupt. 1: Mask 0: Do not mask (default value)
1	PROTM	This bit masks the Protect interrupt. 1: Mask 0: Do not mask (default value)
0	CPUDECM	This bit masks the CPUDEC interrupt. 1: Mask 0: Do not mask (default value)

(18) UF0 INT mask 2 register (UF0IM2)

This register controls masking of the interrupt sources indicated by the UF0IS2 register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7) and the current setting of the interface.

Be sure to clear bits 1 to 3 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM2	BKI2INM	BKI2 DTM	BKI1INM	BKI1 DTM	0	0	0	IT1DTM	00400032H	00H

Bit position	Bit name	Function
7, 5	BKInINM	These bits mask the BLKInIN interrupt. 1: Mask 0: Do not mask (default value)
6, 4	BKInDTM	These bits mask the BLKInDT interrupt. 1: Mask 0: Do not mask (default value)
0	IT1DTM	This bit masks the INT1DT interrupt. 1: Mask 0: Do not mask (default value)

Remark n = 1, 2

(19) UF0 INT mask 3 register (UF0IM3)

This register controls masking of the interrupt sources indicated by the UF0IS3 register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2, 4) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM3	BKO2 FLM	BKO2 NLM	BKO2 NAKM	BKO2 DTM	BKO1 FLM	BKO1 NLM	BKO1 NAKM	BKO1 DTM	00400034H	00H

Bit position	Bit name	Function
7, 3	BKOnFLM	These bits mask the BLKOnFL interrupt. 1: Mask 0: Do not mask (default value)
6, 2	BKOnNLM	These bits mask the BLKOnNL interrupt. 1: Mask 0: Do not mask (default value)
5, 1	BKOnNAKM	These bits mask the BLKOnNK interrupt. 1: Mask 0: Do not mask (default value)
4, 0	BKOnDTM	These bits mask the BLKOnDT interrupt. 1: Mask 0: Do not mask (default value)

Remark n = 1, 2

(20) UF0 INT mask 4 register (UF0IM4)

This register controls masking of the interrupt sources indicated by the UF0IS4 register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1 to 4, 7) and the current setting of the interface.

Be sure to clear bits 0 to 4, 6, and 7 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM4	0	0	SETINTM	0	0	0	0	0	00400036H	00H

Bit position	Bit name	Function
5	SETINTM	This bit masks the SET_INT interrupt. 1: Mask 0: Do not mask (default value)

(21) UF0 INT clear 0 register (UF0IC0)

This register controls clearing the interrupt sources indicated by the UF0IS0 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

Be sure to clear bits 3 to 5 to “1”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC0	BUS RSTC	RSU SPDC	1	1	1	SET RQC	CLR RQC	EP HALTC	0040003CH	FFH

Bit position	Bit name	Function
7	BUSRSTC	This bit clears the Bus Reset interrupt. 0: Clear
6	RSUSPDC	This bit clears the Resume/Suspend interrupt. 0: Clear
2	SETRQC	This bit clears the SET_RQ interrupt. 0: Clear
1	CLRRQC	This bit clears the CLR_RQ interrupt. 0: Clear
0	EPHALTC	This bit clears the EP_Halt interrupt. 0: Clear

(22) UF0 INT clear 1 register (UF0IC1)

This register controls clearing the interrupt sources indicated by the UF0IS1 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC1	1	E0INC	E0 INDTC	E0ODTC	SUCESC	STGC	PROTC	CPU DECC	0040003EH	FFH

Bit position	Bit name	Function
6	E0INC	This bit clears the EP0IN interrupt. 0: Clear
5	E0INDTC	This bit clears the EP0INDT interrupt. 0: Clear
4	E0ODTC	This bit clears the EP0OUTDT interrupt. 0: Clear
3	SUCESC	This bit clears the Success interrupt. 0: Clear
2	STGC	This bit clears the Stg interrupt. 0: Clear
1	PROTC	This bit clears the Protect interrupt. 0: Clear
0	CPUDECC	This bit clears the CPUDEC interrupt. 0: Clear

(23) UF0 INT clear 2 register (UF0IC2)

This register controls clearing the interrupt sources indicated by the UF0IS2 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7) and the current setting of the interface.

Be sure to clear bits 1 to 3 to "1".

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC2	BKI2INC	BKI2 DTC	BKI1INC	BKI1 DTC	1	1	1	IT1DTC	00400040H	FFH

Bit position	Bit name	Function
7, 5	BKInINC	These bits clear the BLKInIN interrupt. 0: Clear
6, 4	BKInDTC	These bits clear the BLKInDT interrupt. 0: Clear
0	IT1DTC	This bit clears the INT1DT interrupt. 0: Clear

Remark n = 1, 2

(24) UF0 INT clear 3 register (UF0IC3)

This register controls clearing the interrupt sources indicated by the UF0IS3 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2, 4) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC3	BKO2 FLC	BKO2 NLC	BKO2 NAKC	BKO2 DTC	BKO1 FLC	BKO1 NLC	BKO1 NAKC	BKO1 DTC	00400042H	FFH

Bit position	Bit name	Function
7, 3	BKOnFLC	These bits clear the BLKOnFL interrupt. 0: Clear
6, 2	BKOnNLC	These bits clear the BLKOnNL interrupt. 0: Clear
5, 1	BKOnNAKC	These bits clear the BLKOnNK interrupt. 0: Clear
4, 0	BKOnDTC	These bits clear the BLKOnDT interrupt. 0: Clear

Remark n = 1, 2

(25) UF0 INT clear 4 register (UF0IC4)

This register controls clearing the interrupt sources indicated by the UF0IS4 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1 to 4, 7) and the current setting of the interface.

Be sure to clear bits 0 to 4, 6, and 7 to “1”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC4	1	1	SETINTC	1	1	1	1	1	00400044H	FFH

Bit position	Bit name	Function
5	SETINTC	This bit clears the SET_INT interrupt. 0: Clear

(26) UF0 FIFO clear 0 register (UF0FIC0)

This register clears each FIFO.

This register is write-only, in 8-bit units. If this register is read, 00H is read.

FW can clear the target FIFO by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7) and the current setting of the interface.

Be sure to clear bit 3 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0FIC0	BKI2SC	BKI2CC	BKI1SC	BKI1CC	0	ITR1C	EP0WC	EP0RC	00400060H	00H

Bit position	Bit name	Function
7, 5	BKInSC	These bits clear only the FIFO on the SIE side of the UF0BIn register (reset the counter). 1: Clear Writing these bits is invalid while an IN token for Endpoint m is being processed with the BKInNK bit set to 1. The BKInNK bit is automatically cleared to 0 by clearing the FIFO. Make sure that the FIFO on the CPU side is empty when these bits are used.
6, 4	BKInCC	These bits clear only the FIFO on the CPU side of the UF0BIn register (reset the counter). 1: Clear
2	ITR1C	This bit clears the UF0INT1 register (reset the counter). 1: Clear Writing this bit is invalid while an IN token for Endpoint7 is being processed with the IT1NK bit set to 1. The IT1NK bit is automatically cleared to 0 by clearing the FIFO.
1	EP0WC	This bit clears the UF0E0W register (resets the counter). 1: Clear Writing this bit is invalid while an IN token for Endpoint0 is being processed with the EP0NKW bit set to 1. The EP0NKW bit is automatically cleared to 0 by clearing the FIFO.
0	EP0RC	This bit clears the UF0E0R register (resets the counter). 1: Clear When the EP0NKR bit is set to 1 (except when it has been set by FW), the EP0NKR bit is automatically cleared to 0 by clearing the FIFO.

Remark n = 1, 2
m = 1 where n = 1
m = 3 where n = 2

(27) UF0 FIFO clear 1 register (UF0FIC1)

This register clears each FIFO.

This register is write-only, in 8-bit units. If this register is read, 00H is read.

FW can clear the target FIFO by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2, 4) and the current setting of the interface.

Be sure to clear bits 4 to 7 to "0".

	7	6	5	4	3	2	1	0	Address	After reset
UF0FIC1	0	0	0	0	BKO2C	BKO2CC	BKO1C	BKO1CC	00400062H	00H

Bit position	Bit name	Function
3, 1	BKOnC	These bits clear the FIFOs on both the SIE and CPU sides of the UF0BOn register (reset the counter). 1: Clear When the BKOnNK bit is set to 1 (except when it has been set by FW), the BKOnNK bit is automatically cleared to 0 by clearing the FIFO.
2, 0	BKOnCC	These bits clear only the FIFO on the CPU side of the UF0BOn register (reset the counter). 1: Clear When the BKOnNK bit is set to 1 (except when it has been set by FW), the BKOnNK bit is automatically cleared to 0 by clearing the FIFO.

Remark n = 1, 2

(28) UF0 data end register (UF0DEND)

This register reports the end of writing to the transmission system.

This register can be read or written in 8-bit units.

FW can start data transfer of the target endpoint by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7) and the current setting of the interface.

Be sure to clear bits 4 and 5 to “0”.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0DEND	BKI2T	BKI1T	0	0	IT1DEND	BKI2DED	BKI1DED	E0DED	0040006AH	00H

Bit position	Bit name	Function
7, 6	BKInT	These bits specify whether toggling the FIFO is automatically executed if the FIFO on the CPU side of the UF0BIn register becomes full. 1: Automatically execute a toggle operation of the FIFO as soon as the FIFO has become full. 0: Do not automatically execute a toggle operation of the FIFO even if the FIFO becomes full (default value).
3	IT1DEND	Set this bit to 1 to transmit the data of the UF0INT1 register. When this bit is set to 1, the IT1NK bit is set to 1 and data transfer is executed. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). If the ITR1C bit of the UF0FIC0 register is set to 1 and then this bit is set to 1 (counter of UF0INT1 register = 0 and the corresponding bit of the UF0EPS0 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0INT1 register and if this bit is set to 1 (counter of UF0INT1 register ≠ 0 and the corresponding bit of the UF0EPS0 register = 1), a short packet is transmitted. This bit is automatically controlled by hardware when the FIFO is full.

Remark n = 1, 2

(2/2)

Bit position	Bit name	Function
2, 1	BKInDED	Set these bits to 1 when writing transmit data to the UF0BIn register has been completed. When these bits are set to 1, the FIFO is toggled as soon as possible, the BKInNK bit is set to 1, and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). These bits control the FIFO on the CPU side. If the BKInCC bit of the UF0FIC0 register is set to 1 and then these bits are set to 1 (counter of UF0BIn register = 0), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0BIn register and if these bits are set to 1 (counter of UF0BIn register ≠ 0), and if the FIFO is not full, a short packet is transmitted. If the FIFO on the CPU side of the UF0BIn register becomes full, with the PIO or BKInT bit set to 1, the hardware starts data transmission even if these bits are not set to 1. If the FIFO on the CPU side of the UF0BIn register becomes full, with the BKInT bit cleared to 0, be sure to set these bits to 1 (see **18.6.3 (3) UF0 EPNAK register (UF0EN)**).
0	E0DED	Set this bit to 1 to transmit data of the UF0E0W register. When this bit is set to 1, the EP0NKW bit is set to 1 and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). If the EP0WC bit of the UF0FIC0 register is set to 1 and if this bit is set to 1 (counter of UF0E0W register = 0 and bit 1 of UF0EPS0 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0E0W register and if this bit is set to 1 (counter of UF0E0W register ≠ 0 and bit 1 of the UF0EPS0 register = 1), and if the FIFO is not full, a short packet is transmitted.

Remark n = 1, 2

(29) UF0 GPR register (UF0GPR)

This register controls USBF and the USB interface.

This register is write-only, in 8-bit units. If this register is read, 00H is read.

FW can reset the USBF by writing 1 to bit 0 of this register. This bit is automatically cleared to 0 after 1 has been written to it. Writing 0 to this bit is invalid.

Be sure to clear bits 1 to 7 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0GPR	0	0	0	0	0	0	0	MRST	0040006EH	00H

Bit position	Bit name	Function
0	MRST	Set this bit to 1 to reset USBF. 1: Reset Actually, USBF is reset two USB clocks after this bit has been set to 1 by FW and the write signal has become inactive. Resetting USBF by the MRST bit while the system clock is operating has the same result as resetting by the $\overline{\text{RESET}}$ pin (hardware reset) (register value back to default value).

(30) UF0 mode control register (UF0MODC)

This register controls CPUDEC processing.

This register can be read or written in 8-bit units.

By setting each bit of this register, the setting of the UF0MODS register can be changed. The bit of this register is automatically cleared to 0 only at hardware reset and when the MRST bit of the UF0GRP register has been set to 1.

Even if the bit of this register has automatically been set to 1 by hardware, the setting by FW takes precedence.

Be sure to clear bits 0 to 5 and 7 to "0". If they are set to 1, the operation is not guaranteed.

Caution This register is provided for debugging purposes. Usually, do not set this register except for verifying the operation or when a special mode is used.

	7	6	5	4	3	2	1	0	Address	After reset
UF0MODC	0	CDC GDST	0	0	0	0	0	0	00400074H	00H

Bit position	Bit name	Function
6	CDCGDST	Set this bit to 1 to switch the GET_DESCRIPTOR Configuration request to CPUDEC processing. By setting this bit to 1, the CDCGD bit of the UF0MODS register can be forcibly set to 1. 1: Forcibly change the GET_DESCRIPTOR Configuration request to CPUDEC processing (sets the CDCGD bit of the UF0MODS register to 1). 0: Automatically process the GET_DESCRIPTOR Configuration request (default value).

(31) UF0 mode status register (UF0MODS)

This register indicates the configuration status.

This register is read-only, in 8-bit units.

Be sure to clear bits 0, 1, 5, and 7 to “0”.

	7	6	5	4	3	2	1	0	Address	After reset
UF0MODS	0	CDCGD	0	MPACK	DFLT	CONF	0	0	00400078H	00H

Bit position	Bit name	Function
6	CDCGD	This bit specifies whether CPUDEC processing is performed for the GET_DESCRIPTOR Configuration request. 1: Forcibly change the GET_DESCRIPTOR Configuration request to CPUDEC processing. 0: Automatically process the GET_DESCRIPTOR Configuration request (default value).
4	MPACK	This bit indicates the transmit packet size of Endpoint0. 1: Transmit a packet of other than 8 bytes. 0: Transmit a packet of 8 bytes (default value). This bit is automatically set to 1 by hardware after the GET_DESCRIPTOR Device request has been processed (on normal completion of the status stage). It is not cleared to 0 until the USBF has been reset (it is not cleared to 0 by Bus Reset). If this bit is not set to 1, the hardware transfers only the automatically-executed request in 8-byte units. Therefore, even if data of more than 8 bytes is sent by the OUT token to be processed by FW before completion of the GET_DESCRIPTOR Device request, the data is correctly received. This bit is ignored if the size of Endpoint0 is 8 bytes.
3	DFLT	This bit indicates the default status (DFLT bit = 1). 1: Enables response. 0: Disables response (always no response) (default value). This bit is automatically set to 1 by Bus Reset. The transaction for all the endpoints is not responded to until this bit is set to 1.
2	CONF	This bit indicates whether the SET_CONFIGURATION request has been completed. 1: SET_CONFIGURATION request has been completed. 0: SET_CONFIGURATION request has not been completed (default value). This bit is set to 1 when Configuration value = 1 is received by the SET_CONFIGURATION request. Unless this bit is set to 1, access to an endpoint other than Endpoint0 is ignored. This bit is cleared to 0 when Configuration value = 0 is received by the SET_CONFIGURATION request. It is also cleared to 0 when Bus Reset is detected.

(32) UF0 active interface number register (UF0AIFN)

This register sets the valid Interface number that correctly responds to the GET/SET_INTERFACE request. Because Interface 0 is always valid, Interfaces 1 to 4 can be selected.

This register can be read or written in 8-bit units.

Be sure to clear bits 0, 1, 5, and 7 to "0".

	7	6	5	4	3	2	1	0	Address	After reset
UF0AIFN	ADDIF	0	0	0	0	0	IFNO1	IFNO0	00400080H	00H

Bit position	Bit name	Function
7	ADDIF	This bit allows use of Interfaces numbered other than 0. 1: Support up to the Interface number specified by the IFNO1 and IFNO0 bits. 0: Support only Interface 0 (default value). Setting bits 1 and 0 of this register is invalid when this bit is not set to 1.
1, 0	IFNO1, IFNO0	These bits specify the range of Interface numbers to be supported.

IFNO1	IFNO0	Valid Interface No.
1	1	0, 1, 2, 3, 4
1	0	0, 1, 2, 3
0	1	0, 1, 2
0	0	0, 1

(33) UF0 active alternative setting register (UF0AAS)

This register specifies a link between the Interface number and Alternative Setting.

This register can be read or written in 8-bit units.

USBF of the V850E/IG4-H and V850E/IH4-H can set a five-series Alternative Setting (Alternate Setting 0, 1, 2, 3, and 4 can be defined) and a two-series Alternative Setting (Alternative Setting 0 and 1 can be defined) for one Interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0AAS	ALT2	IFAL21	IFAL20	ALT2EN	ALT5	IFAL51	IFAL50	ALT5EN	00400082H	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 3</td><td>ALTn</td><td>These bits specify whether an n-series Alternative Setting is linked with Interface 0. When these bits are set to 1, the setting of the IFALn1 and IFALn0 bits is invalid.
1: Link n-series Alternative Setting with Interface 0.
0: Do not link n-series Alternative Setting with Interface 0 (default value).</td></tr>
<tr><td>6, 5,
2, 1</td><td>IFALn1,
IFALn0</td><td>These bits specify the Interface number to be linked with the n-series Alternative Setting. If the linked Interface number is outside the range specified by the UF0AIFN register, the n-series Alternative Setting is invalid (ALTnEN bit = 0).
<table>
<tr><th>IFALn1</th><th>IFALn0</th><th>Interface number to be linked</th></tr>
<tr><td>1</td><td>1</td><td>Links Interface 4.</td></tr>
<tr><td>1</td><td>0</td><td>Links Interface 3.</td></tr>
<tr><td>0</td><td>1</td><td>Links Interface 2.</td></tr>
<tr><td>0</td><td>0</td><td>Links Interface 1.</td></tr>
</table>
Do not link a five-series Alternative Setting and a two-series Alternative Setting with the same Interface number.</td></tr>
<tr><td>4, 0</td><td>ALTnEN</td><td>These bits validate the n-series Alternative Setting. Unless these bits are set to 1, the setting of the ALTn, IFALn1, and IFALn0 bits is invalid.
1: Validate the n-series Alternative Setting.
0: Do not validate the n-series Alternative Setting (default value).</td></tr>
</table>

Remark n = 2, 5

For example, when the UF0AIFN register is set to 82H and the UF0AAS register is set to 15H, Interfaces 0, 1, 2, and 3 are valid. Interfaces 0 and 2 support only Alternative Setting 0. Interface 1 supports Alternative Setting 0 and 1, and Interface 3 supports Alternative Setting 0, 1, 2, 3, and 4. With this setting, requests GET_INTERFACE wIndex = 0/1/2/3, SET_INTERFACE wValue = 0 & wIndex = 0/2, SET_INTERFACE wValue = 0/1 & wIndex = 1, and SET_INTERFACE wValue = 0/1/2/3/4 & wIndex = 3 are automatically responded to, and a STALL response is made to the other GET/SET_INTERFACE requests.

(34) UF0 alternative setting status register (UF0ASS)

This register indicates the current status of the Alternative Setting.

This register is read-only, in 8-bit units.

Check this register when the SET_INT interrupt request has been issued. The value received by the SET_INTERFACE request is reflected on the UF0IFn register (n = 0 to 4) as well as on this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ASS	0	0	0	0	AL5ST3	AL5ST2	AL5ST1	AL2ST	00400084H	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 1</td><td>AL5ST3 to AL5ST1</td><td>These bits indicate the current status of the five-series Alternative Setting.
<table>
<tr><th>AL5ST3</th><th>AL5ST2</th><th>AL5ST1</th><th>Selected Alternative Setting number</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Alternative Setting 4</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Alternative Setting 3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Alternative Setting 2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Alternative Setting 1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Alternative Setting 0</td></tr>
</table></td></tr>
<tr><td>0</td><td>AL2ST</td><td>This bit indicates the current status of the two-series Alternative Setting (selected Alternative Setting number).
1: Alternative Setting 1
0: Alternative Setting 0</td></tr>
</table>

(35) UF0 endpoint 1 interface mapping register (UF0E1IM)

This register specifies for which Interface and Alternative Setting Endpoint1 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint1 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint1 request and the IN transaction to Endpoint1 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E1IM	E1EN2	E1EN1	E1EN0	E12AL1	E15AL4	E15AL3	E15AL2	E15AL1	00400086H	00H

Bit position	Bit name	Function
7 to 5	E1EN2 to E1EN0	These bits set a link between the Interface of Endpoint1 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E1EN2	E1EN1	E1EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternative Setting 0
1	0	0	Linked with Interface 3 and Alternative Setting 0
0	1	1	Linked with Interface 2 and Alternative Setting 0
0	1	0	Linked with Interface 1 and Alternative Setting 0
0	0	1	Linked with Interface 0 and Alternative Setting 0
0	0	0	Not linked with Interface (default value)

Bit position	Bit name	Function
		When these bits are set to 110 or 111, they are invalid even if the E12AL1 bit is cleared to 0. If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint1 is valid.
4	E12AL1	This bit validates Endpoint1 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E15AL4 to E15AL1 bits are 0000.
3 to 0	E15ALn	These bits validate Endpoint1 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).

Remark n = 1 to 4

(36) UF0 endpoint 2 interface mapping register (UF0E2IM)

This register specifies for which Interface and Alternative Setting Endpoint2 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint2 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint2 request and the OUT transaction to Endpoint2 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E2IM	E2EN2	E2EN1	E2EN0	E22AL1	E25AL4	E25AL3	E25AL2	E25AL1	00400088H	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7 to 5</td><td>E2EN2 to
E2EN0</td><td>These bits set a link between the Interface of Endpoint2 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.
<table>
<tr><th>E2EN2</th><th>E2EN1</th><th>E2EN0</th><th>Link status</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Not linked with Interface</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Linked with Interface 4 and Alternative Setting 0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Linked with Interface 3 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Linked with Interface 2 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Linked with Interface 1 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Linked with Interface 0 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Not linked with Interface (default value)</td></tr>
</table>
When these bits are set to 110 or 111, they are invalid even if the E22AL1 bit is cleared to 0.
If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint2 is valid.</td></tr>
<tr><td>4</td><td>E22AL1</td><td>This bit validates Endpoint2 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1.
1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1.
0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value).
This bit is valid when the E25AL4 to E25AL1 bits are 0000.</td></tr>
<tr><td>3 to 0</td><td>E25ALn</td><td>These bits validate Endpoint2 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n.
1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1.
0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).</td></tr>
</table>

Remark n = 1 to 4

(37) UF0 endpoint 3 interface mapping register (UF0E3IM)

This register specifies for which Interface and Alternative Setting Endpoint3 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint3 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint3 request and the IN transaction to Endpoint3 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E3IM	E3EN2	E3EN1	E3EN0	E32AL1	E35AL4	E35AL3	E35AL2	E35AL1	0040008AH	00H

Bit position	Bit name	Function
7 to 5	E3EN2 to E3EN0	These bits set a link between the Interface of Endpoint3 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E3EN2	E3EN1	E3EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternative Setting 0
1	0	0	Linked with Interface 3 and Alternative Setting 0
0	1	1	Linked with Interface 2 and Alternative Setting 0
0	1	0	Linked with Interface 1 and Alternative Setting 0
0	0	1	Linked with Interface 0 and Alternative Setting 0
0	0	0	Not linked with Interface (default value)

Bit position	Bit name	Function
7 to 5	E3EN2 to E3EN0	When these bits are set to 110 or 111, they are invalid even if the E32AL1 bit is cleared to 0. If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint3 is valid.
4	E32AL1	This bit validates Endpoint3 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E35AL4 to E35AL1 bits are 0000.
3 to 0	E35ALn	These bits validate Endpoint3 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).

Remark n = 1 to 4

(38) UF0 endpoint 4 interface mapping register (UF0E4IM)

This register specifies for which Interface and Alternative Setting Endpoint4 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint4 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint4 request and the OUT transaction to Endpoint4 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E4IM	E4EN2	E4EN1	E4EN0	E42AL1	E45AL4	E45AL3	E45AL2	E45AL1	0040008CH	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7 to 5</td><td>E4EN2 to E4EN0</td><td>These bits set a link between the Interface of Endpoint4 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.
<table>
<tr><th>E4EN2</th><th>E4EN1</th><th>E4EN0</th><th>Link status</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td rowspan="2">Not linked with Interface</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Linked with Interface 4 and Alternative Setting 0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Linked with Interface 3 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Linked with Interface 2 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Linked with Interface 1 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Linked with Interface 0 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Not linked with Interface (default value)</td></tr>
</table>
When these bits are set to 110 or 111, they are invalid even if the E42AL1 bit is cleared to 0.
If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint4 is valid.</td></tr>
<tr><td>4</td><td>E42AL1</td><td>This bit validates Endpoint4 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1.
1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1.
0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value).
This bit is valid when the E45AL4 to E45AL1 bits are 0000.</td></tr>
<tr><td>3 to 0</td><td>E45ALn</td><td>These bits validate Endpoint4 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n.
1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1.
0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).</td></tr>
</table>

Remark n = 1 to 4

(39) UF0 endpoint 7 interface mapping register (UF0E7IM)

This register specifies for which Interface and Alternative Setting Endpoint7 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint7 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint7 request and the IN transaction to Endpoint7 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E7IM	E7EN2	E7EN1	E7EN0	E72AL1	E75AL4	E75AL3	E75AL2	E75AL1	00400092H	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7 to 5</td><td>E7EN2 to E7EN0</td><td>These bits set a link between the Interface of Endpoint7 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.
<table>
<tr><th>E7EN2</th><th>E7EN1</th><th>E7EN0</th><th>Link status</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Not linked with Interface</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Linked with Interface 4 and Alternative Setting 0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Linked with Interface 3 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Linked with Interface 2 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Linked with Interface 1 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Linked with Interface 0 and Alternative Setting 0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Not linked with Interface (default value)</td></tr>
</table>
When these bits are set to 110 or 111, they are invalid even if the E72AL1 bit is cleared to 0.
If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint7 is valid.</td></tr>
<tr><td>4</td><td>E72AL1</td><td>This bit validates Endpoint7 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1.
1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1.
0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value).
This bit is valid when the E75AL4 to E75AL1 bits are 0000.</td></tr>
<tr><td>3 to 0</td><td>E75ALn</td><td>These bits validate Endpoint7 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n.
1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1.
0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).</td></tr>
</table>

Remark n = 1 to 4

18.6.4 Data hold registers

(1) UF0 EP0 read register (UF0E0R)

The UF0E0R register is a 64-byte FIFO that stores the OUT data sent from the host in the data stage of control transfer to/from Endpoint0.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The hardware automatically transfers data to the UF0E0R register when it has received the data from the host. When the data has been correctly received, the E0ODT bit of the UF0IS1 register is set to 1. The UF0E0L register holds the quantity of the received data, and an interrupt request (INTUSBF0) is issued. The UF0E0L register always updates the length of the received data while it is receiving data. If the final transfer is correct reception, the interrupt request is generated. If the reception is abnormal, the UF0E0L register is cleared to 0 and the interrupt request is not generated.

The data held by the UF0E0R register must be read by FW up to the value of the amount of data read by the UF0E0L register. Check that all data has been read by using the EP0R bit of the UF0EPS0 register (EP0R bit = 0 when all data has been read). If the value of the UF0E0L register is 0, the EP0NKR bit of the UF0E0N register is cleared to 0, and the UF0E0R register is ready for reception. The UF0E0R register is cleared when the next SETUP token has been received.

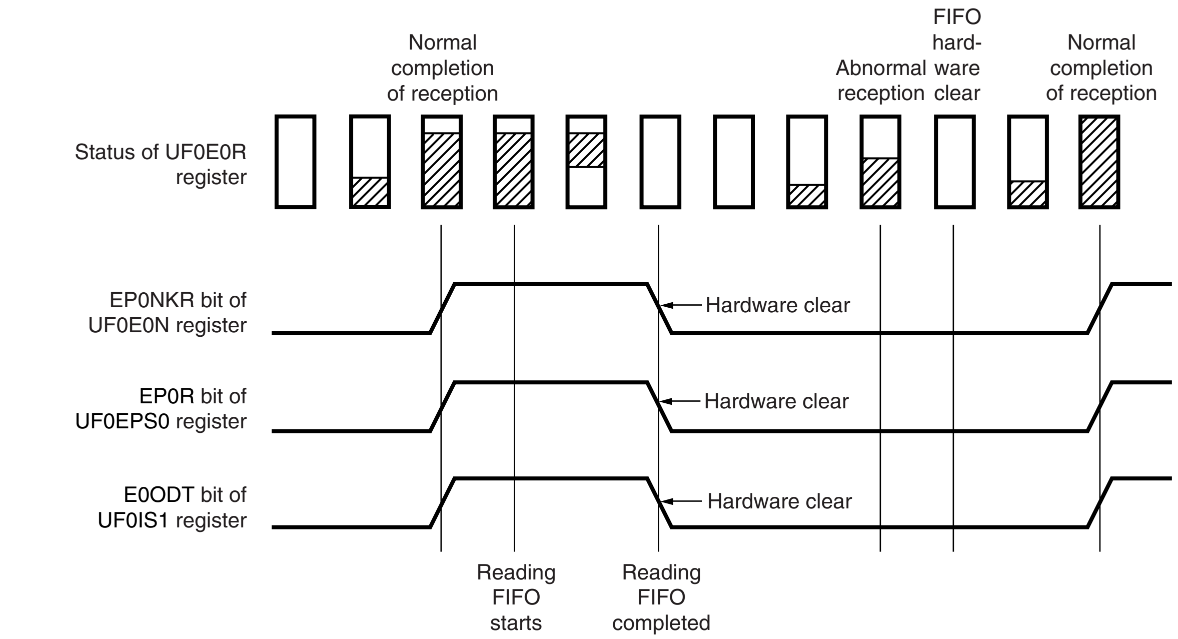
Caution Read all the data stored. Clear the FIFO to discard some data.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0R	E0R7	E0R6	E0R5	E0R4	E0R3	E0R2	E0R1	E0R0	00400100H	Undefined

Bit position	Bit name	Function
7 to 0	E0R7 to E0R0	These bits store the OUT data sent from the host in the data stage of control transfer to/from Endpoint0.

The operation of the UF0E0R register is illustrated below.

Figure 18-4. Operation of UF0E0R Register


Normal completion of reception
Abnormal reception
FIFO hard-ware clear
Normal completion of reception
Status of UF0E0R register
EP0NKR bit of UF0E0N register
Hardware clear
EP0R bit of UF0EPS0 register
Hardware clear
E0ODT bit of UF0IS1 register
Hardware clear
Reading FIFO starts
Reading FIFO completed

(2) UF0 EP0 length register (UF0E0L)

The UF0E0L register stores the data length held by the UF0E0R register.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0E0L register always updates the length of the received data while it is receiving data. If the final transfer is abnormal reception, the UF0E0L register is cleared to 0 and the interrupt request is not generated. The interrupt request is generated only when the reception is normal, and the FW can read as many data from the UF0E0R register as the value read from the UF0E0L register. The value of the UF0E0L register is decremented each time the UF0E0R register has been read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0L	E0L7	E0L6	E0L5	E0L4	E0L3	E0L2	E0L1	E0L0	00400102H	00H

Bit position	Bit name	Function
7 to 0	E0L7 to E0L0	These bits store the data length held by the UF0E0R register.

(3) UF0 EP0 setup register (UF0E0ST)

The UF0E0ST register holds the SETUP data sent from the host.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0E0ST register always writes data when a SETUP transaction has been received. The hardware sets the PROT bit of the UF0IS1 register when it has correctly received the SETUP transaction. It sets the CPUDEC bit of the UF0IS1 register in the case of an FW-processed request. Then an interrupt request (INTUSBF0) is issued. In the case of an FW-processed request, be sure to read the request in 8-byte units. If it is not read in 8-byte units, the subsequent requests cannot be correctly decoded. The read counter of the UF0E0ST register is not cleared even when Bus Reset is received. Always read this counter in 8-byte units regardless of whether Bus Reset is received or not.

Because the UF0E0ST register always enables writing, the hardware overwrites data to this register even if a SETUP transaction is received while the data of the register is being read. Even if the SETUP transaction cannot be correctly received, the CPUDEC interrupt request and Protect interrupt request are not generated, but the previous data is discarded. If a SETUP token of less than 8 bytes is received, however, the received SETUP token is discarded, and the previously received SETUP data is retained. If the SETUP token is received more than once when control transfer is executed once, be sure to check the PROT bit of the UF0IS1 register under the conditions below. If PROT bit = 1, read the UF0E0ST register again because the SETUP transaction has been received more than once.

<1> If a request is decoded by FW and the UF0E0R register is read or the UF0E0W register is written

<2> When preparing for a STALL response for the request to which the decode result does not correspond

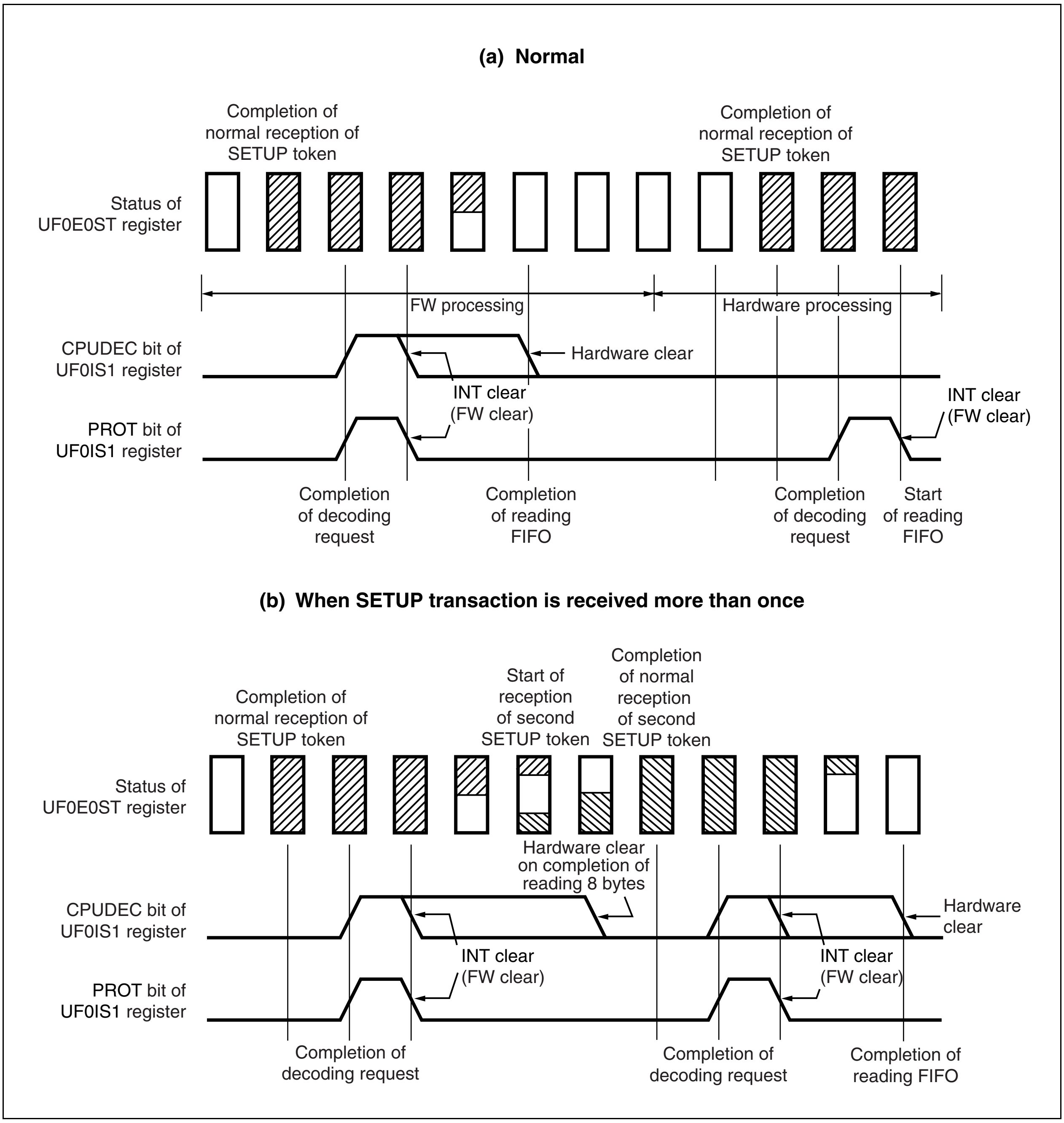
Caution Be sure to read all the stored data. The UF0E0ST register is always updated by the request in the SETUP transaction.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0ST	E0S7	E0S6	E0S5	E0S4	E0S3	E0S2	E0S1	E0S0	00400104H	00H

Bit position	Bit name	Function
7 to 0	E0S7 to E0S0	These bits hold the SETUP data sent from the host.

The operation of the UF0E0ST register is illustrated below.

Figure 18-5. Operation of UF0E0ST Register


(a) Normal
Completion of normal reception of SETUP token
Completion of normal reception of SETUP token
Status of UF0E0ST register
FW processing
Hardware processing
CPUDEC bit of UF0IS1 register
Hardware clear
INT clear (FW clear)
INT clear (FW clear)
PROT bit of UF0IS1 register
Completion of decoding request
Completion of reading FIFO
Completion of decoding request
Start of reading FIFO
(b) When SETUP transaction is received more than once
Completion of normal reception of SETUP token
Start of reception of second SETUP token
Completion of normal reception of second SETUP token
Status of UF0E0ST register
Hardware clear on completion of reading 8 bytes
CPUDEC bit of UF0IS1 register
Hardware clear
INT clear (FW clear)
INT clear (FW clear)
PROT bit of UF0IS1 register
Completion of decoding request
Completion of decoding request
Completion of reading FIFO

(4) UF0 EP0 write register (UF0E0W)

The UF0E0W register is a 64-byte FIFO that stores the IN data (passes it to SIE) sent to the host in the data stage to Endpoint0.

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with an IN token only when the EP0NKW bit of the UF0E0N register is set to 1 (when NAK is not transmitted). When data is transmitted and when the host correctly receives the data, the EP0NKW bit of the UF0E0N register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the UF0E0W register and the E0DED bit of the UF0DEND register is set to 1 (EP0W bit of the UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0E0W register is cleared and the E0DED bit of the UF0DEND register is set to 1 (EP0W bit of the UF0EPS0 register = 1 (data exists)).

The UF0E0W register is cleared to 0 when the next SETUP token is received while transmission has not been completed yet. If the stage of control transfer (read) changes to the status stage while ACK has not been correctly received in the data stage, the UF0E0W register is automatically cleared to 0. At the same time, it is also cleared to 0 if the EP0NKW bit of the UF0E0N register is 1.

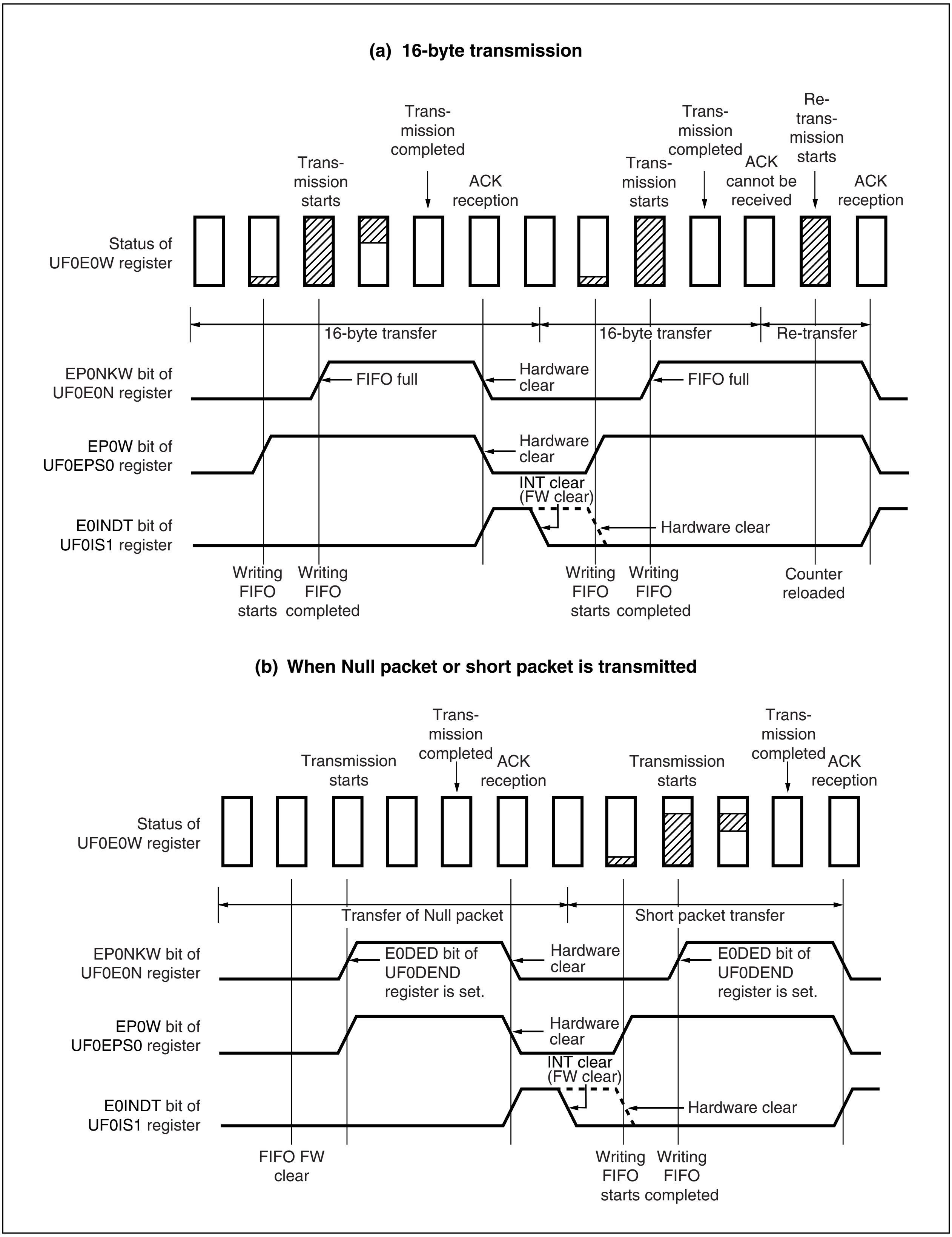
If the UF0E0W register is read while no data is in it, 00H is read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0W	E0W7	E0W6	E0W5	E0W4	E0W3	E0W2	E0W1	E0W0	00400106H	Undefined

Bit position	Bit name	Function
7 to 0	E0W7 to E0W0	These bits store the IN data sent to the host in the data stage to Endpoint0.

The operation of the UF0E0W register is illustrated below.

Figure 18-6. Operation of UF0E0W Register



(5) UF0 bulk-out 1 register (UF0BO1)

The UF0BO1 register is a 64-byte × 2 FIFO that stores data for Endpoint2. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when data is in the FIFO on the SIE side and when no data is in the FIFO on the CPU side (counter value = 0).

This register is read-only, in 8-bit units. A write access to this register is ignored.

When the hardware receives data for Endpoint2 from the host, it automatically transfers the data to the UF0BO1 register. When the register correctly receives the data, a FIFO toggle operation occurs. As a result, the BKO1DT bit of the UF0IS3 register is set to 1, the quantity of the received data is held by the UF0BO1L register, and an interrupt request is issued to the CPU.

Read the data held by the UF0BO1 register by FW, up to the value of the amount of data read by the UF0BO1L register. When the correct received data is held by the FIFO connected to the SIE side and the value of the UF0BO1L register reaches 0, the toggle operation of the FIFO occurs, and the BKO1NK bit of the UF0EN register is automatically cleared to 0. If data greater than the value of the UF0BO1L register is read and if the FIFO toggle condition is satisfied, the toggle operation of the FIFO occurs. As a result, the next packet may be read by mistake. Note that, if the toggle condition is not satisfied, the first data is repeatedly read.

If overrun data is received while data is held by the FIFO connected to the CPU side, Endpoint2 stalls, and the FIFO on the CPU side is cleared.

When the UF0BO1 register is read while no data is in it, an undefined value is read.

Caution Be sure to read all the data stored in this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO1	BKO17	BKO16	BKO15	BKO14	BKO13	BKO12	BKO11	BKO10	00400108H	Undefined

Bit position	Bit name	Function
7 to 0	BKO17 to BKO10	These bits store data for Endpoint2.

The operation of the UF0BO1 register is illustrated below.

Figure 18-7. Operation of UF0BO1 Register (1/2)

(a) Operation example 1

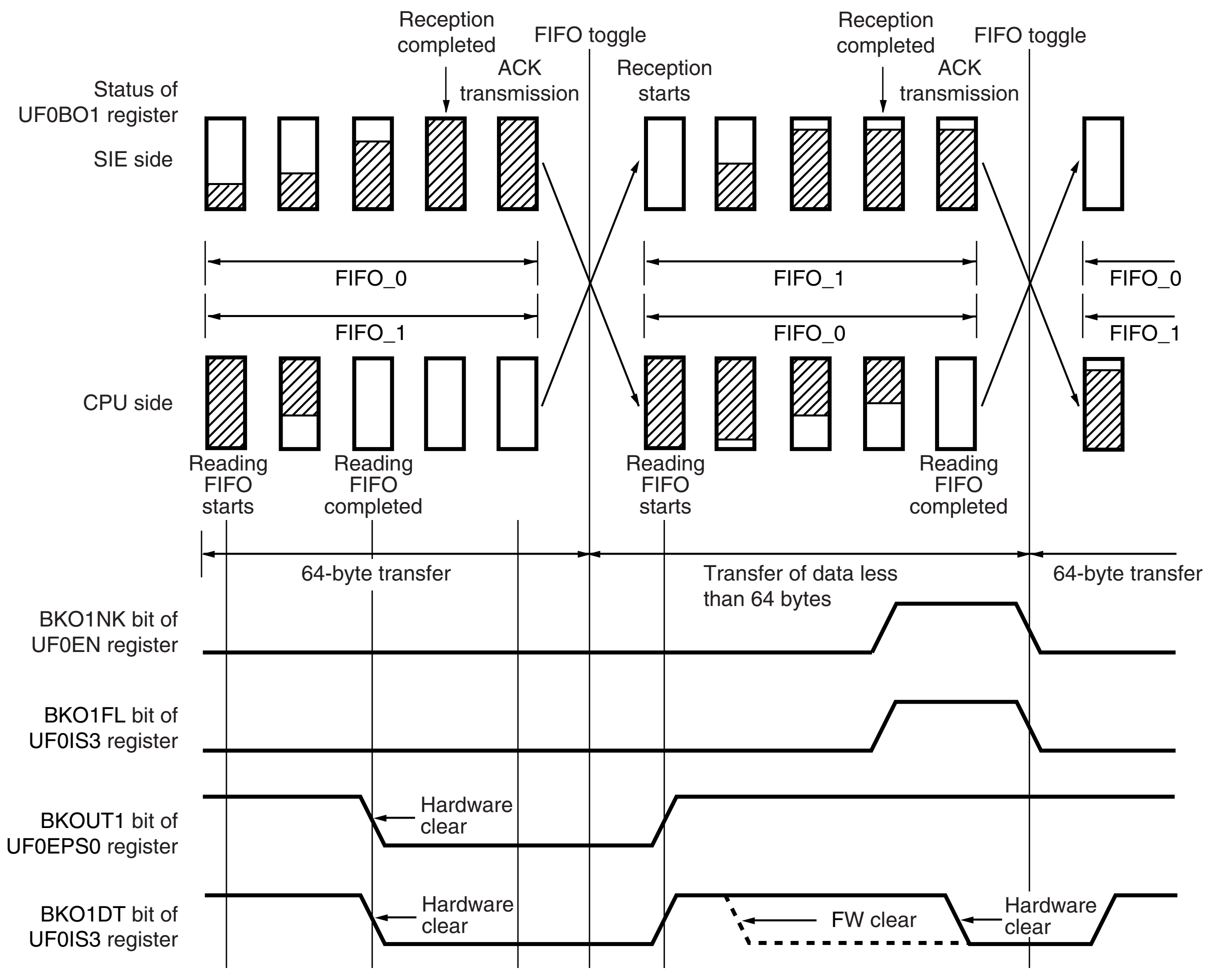
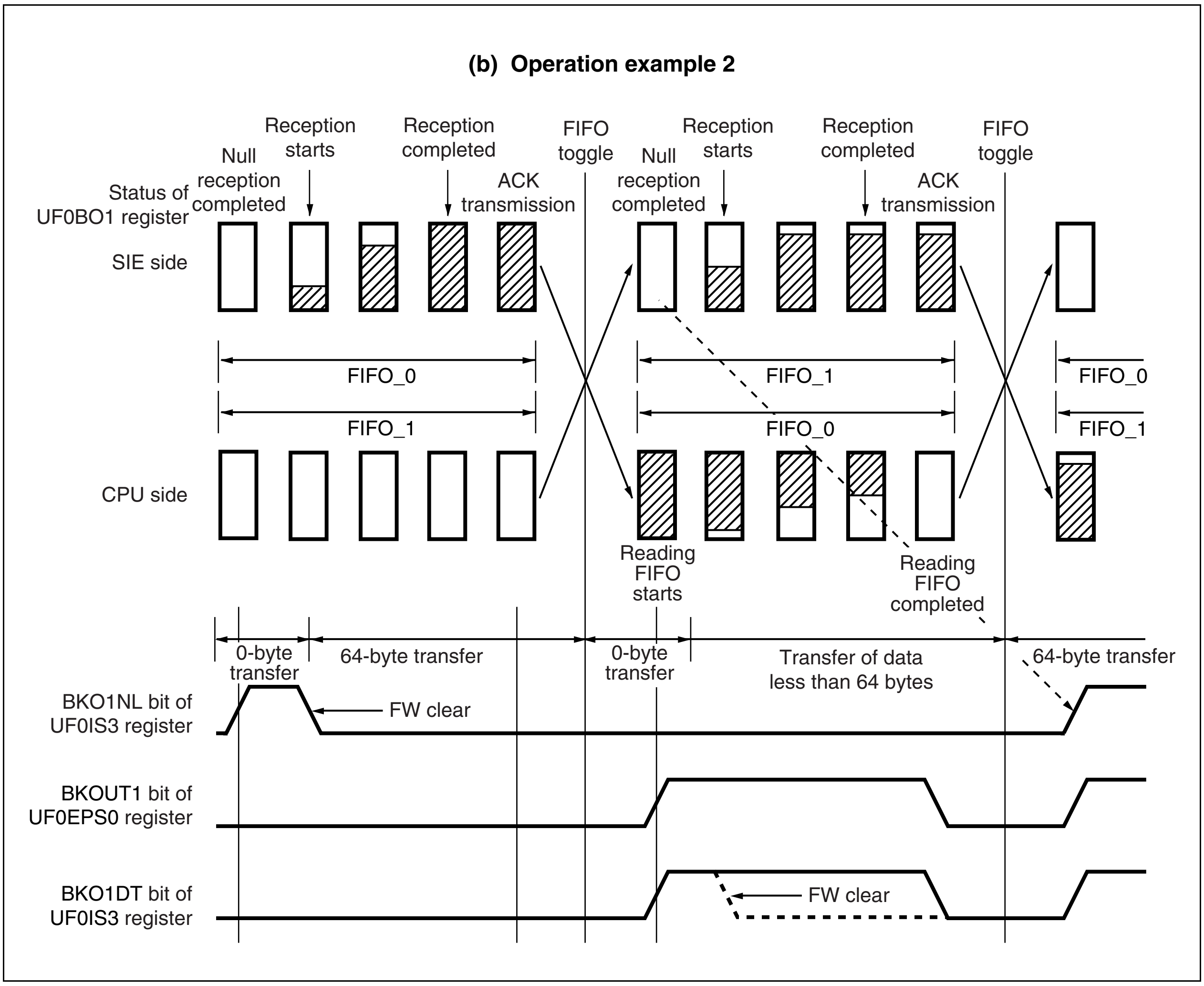


Figure 18-7. Operation of UF0BO1 Register (2/2)



(6) UF0 bulk-out 1 length register (UF0BO1L)

The UF0BO1L register stores the length of the data held by the UF0BO1 register.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0BO1L register always updates the received data length while it is receiving data. If the final transfer is abnormal reception, the UF0BO1L register is cleared to 00H, and an interrupt request is not generated. Only if the reception is normal, the interrupt request is generated, and FW can read as much data from the UF0BO1 register as the value read from the UF0BO1L register. The value of the UF0BO1L register is decremented each time the UF0BO1 register has been read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO1L	BKO1L7	BKO1L6	BKO1L5	BKO1L4	BKO1L3	BKO1L2	BKO1L1	BKO1L0	0040010AH	00H

Bit position	Bit name	Function
7 to 0	BKO1L7 to BKO1L0	These bits store the length of the data held by the UF0BO1 register.

(7) UF0 bulk-out 2 register (UF0BO2)

The UF0BO2 register is a 64-byte × 2 FIFO that stores data for Endpoint4. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when data is in the FIFO on the SIE side and when no data is in the FIFO on the CPU side (counter value = 0).

This register is read-only, in 8-bit units. A write access to this register is ignored.

When the hardware receives data for Endpoint4 from the host, it automatically transfers the data to the UF0BO2 register. When the register correctly receives the data, a FIFO toggle operation occurs. As a result, the BKO2DT bit of the UF0IS3 register is set to 1, the quantity of the received data is held by the UF0BO2L register, and an interrupt request is issued to the CPU.

Read the data held by the UF0BO2 register by FW, up to the value of the amount of data read by the UF0BO2L register. When the correct received data is held by the FIFO connected to the SIE side and the value of the UF0BO2L register reaches 0, the toggle operation of the FIFO occurs, and the BKO2NK bit of the UF0EN register is automatically cleared to 0. If data greater than the value of the UF0BO2L register is read and if the FIFO toggle condition is satisfied, the toggle operation of the FIFO occurs. As a result, the next packet may be read by mistake. Note that, if the toggle condition is not satisfied, the first data is repeatedly read.

If overrun data is received while data is held by the FIFO connected to the CPU side, Endpoint4 stalls, and the FIFO on the CPU side is cleared.

When the UF0BO2 register is read while no data is in it, an undefined value is read.

Caution Be sure to read all the data stored in this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO2	BKO27	BKO26	BKO25	BKO24	BKO23	BKO22	BKO21	BKO20	0040010CH	Undefined

Bit position	Bit name	Function
7 to 0	BKO27 to BKO20	These bits store data for Endpoint4.

The operation of the UF0BO2 register is illustrated below.

Figure 18-8. Operation of UF0BO2 Register (1/2)

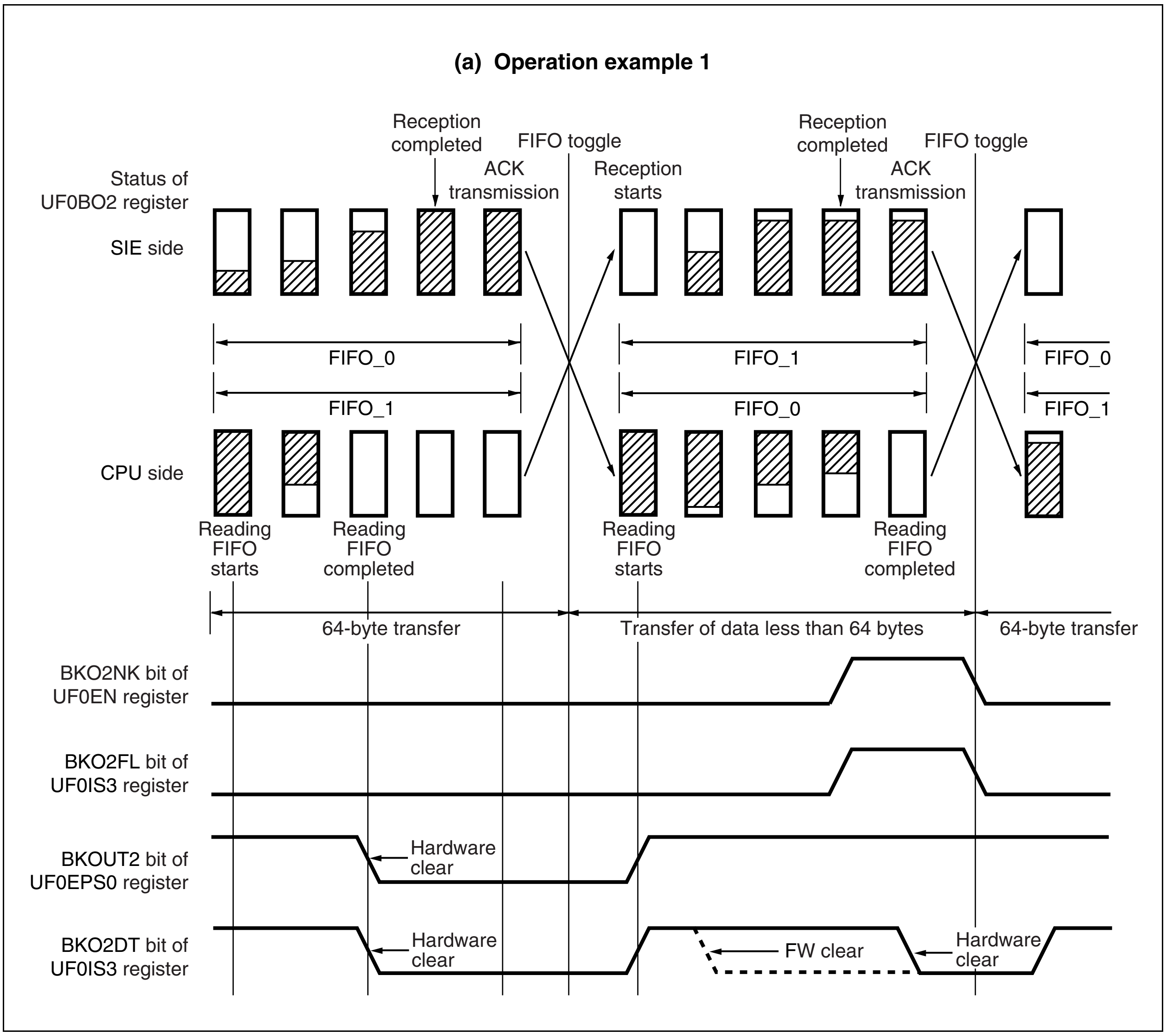
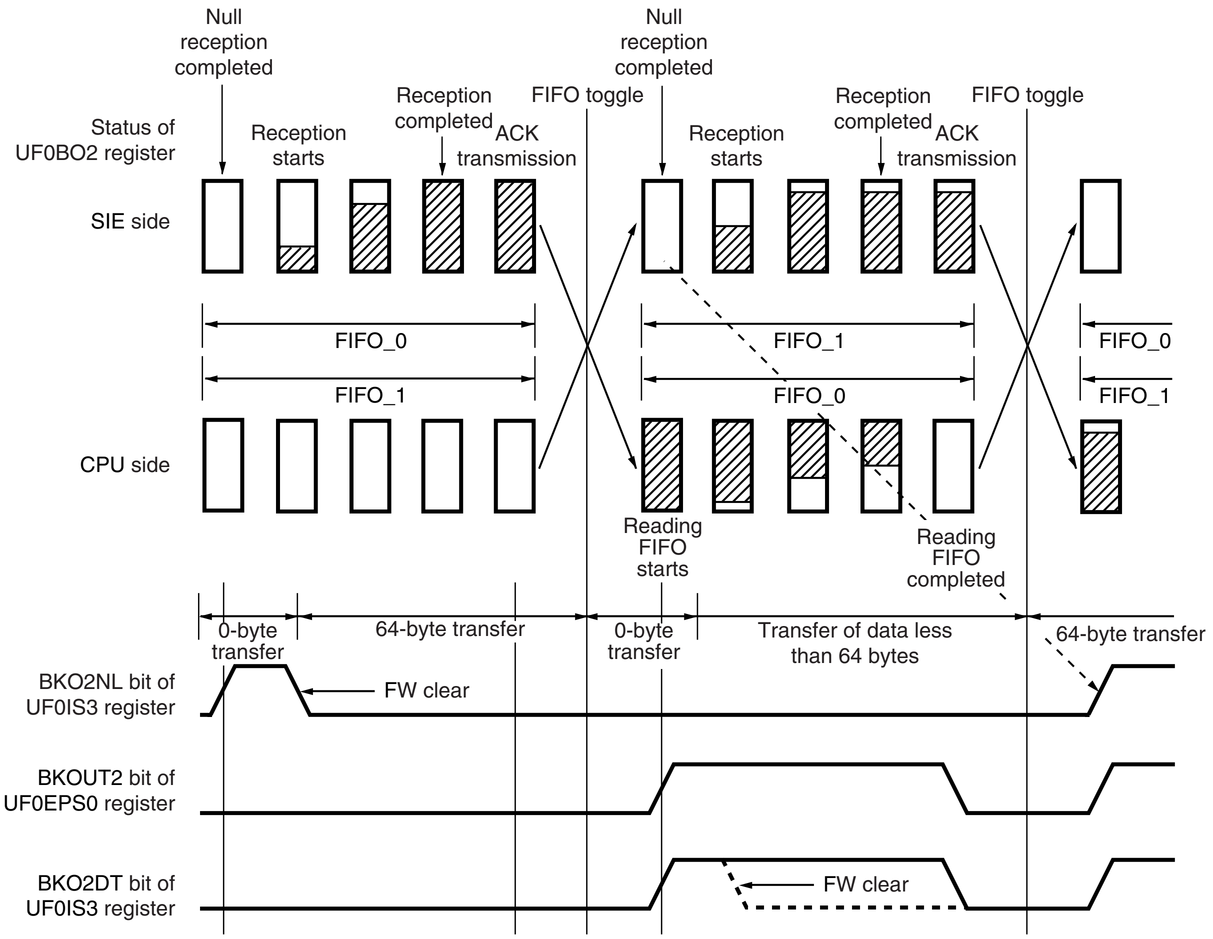


Figure 18-8. Operation of UF0BO2 Register (2/2)

(b) Operation example 2



(8) UF0 bulk-out 2 length register (UF0BO2L)

The UF0BO2L register stores the length of the data held by the UF0BO2 register.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0BO2L register always updates the received data length while it is receiving data. If the final transfer is abnormal reception, the UF0BO2L register is cleared to 00H, and an interrupt request is not generated. Only if the reception is normal, the interrupt request is generated, and FW can read as much data from the UF0BO2 register as the value read from the UF0BO2L register. The value of the UF0BO2L register is decremented each time the UF0BO2 register has been read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO2L	BKO2L7	BKO2L6	BKO2L5	BKO2L4	BKO2L3	BKO2L2	BKO2L1	BKO2L0	0040010EH	00H

Bit position	Bit name	Function
7 to 0	BKO2L7 to BKO2L0	These bits store the length of the data held by the UF0BO2 register.

(9) UF0 bulk-in 1 register (UF0BI1)

The UF0BI1 register is a 64-byte × 2 FIFO that stores data for Endpoint1. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when no data is in the FIFO on the SIE side (counter value = 0) and when the FIFO on the CPU side is correctly written (FIFO full or BKI1DED bit = 1).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint1 only when the BKI1NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). The address at which data is to be written or read is managed by the hardware. Therefore, FW can transmit data to the host only by writing the data to the UF0BI1 register sequentially. A short packet is transmitted when data is written to the UF0BI1 register and the BKI1DED bit of the UF0DEND register is set to 1 (BKIN1 bit of UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0BI1 register is cleared and the BKI1DED bit of the UF0DEND register is set to 1 (BKIN1 bit of the UF0EPS0 register = 1 (data exists)). When the data is transmitted correctly, a FIFO toggle operation occurs. The BKI1DT bit of the UF0IS2 register is set to 1, and an interrupt request is generated for the CPU.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BI1	BKI17	BKI16	BKI15	BKI14	BKI13	BKI12	BKI11	BKI10	00400110H	Undefined

Bit position	Bit name	Function
7 to 0	BKI17 to BKI10	These bits store data for Endpoint1.

The operation of the UF0BI1 register is illustrated below.

Figure 18-9. Operation of UF0BI1 Register (1/3)

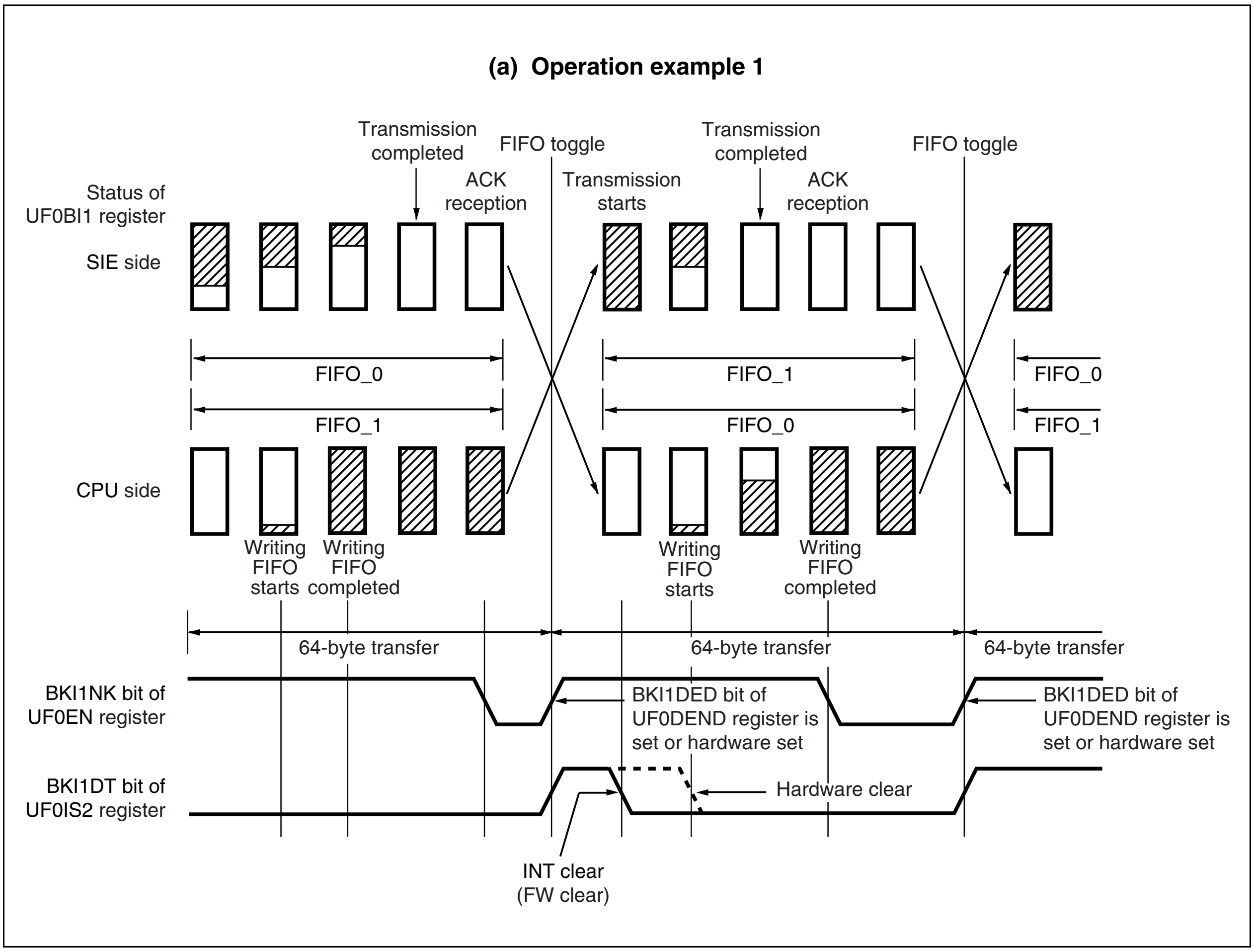


Figure 18-9. Operation of UF0BI1 Register (2/3)

(b) Operation example 2

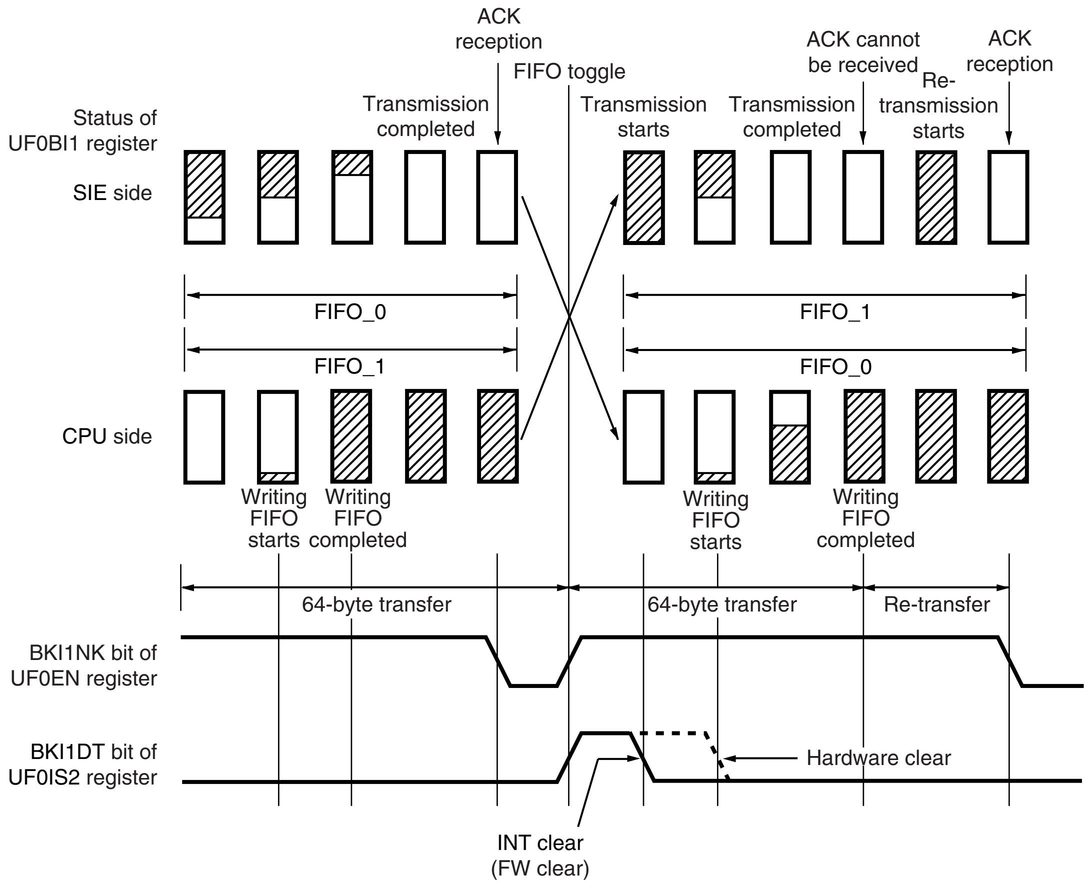
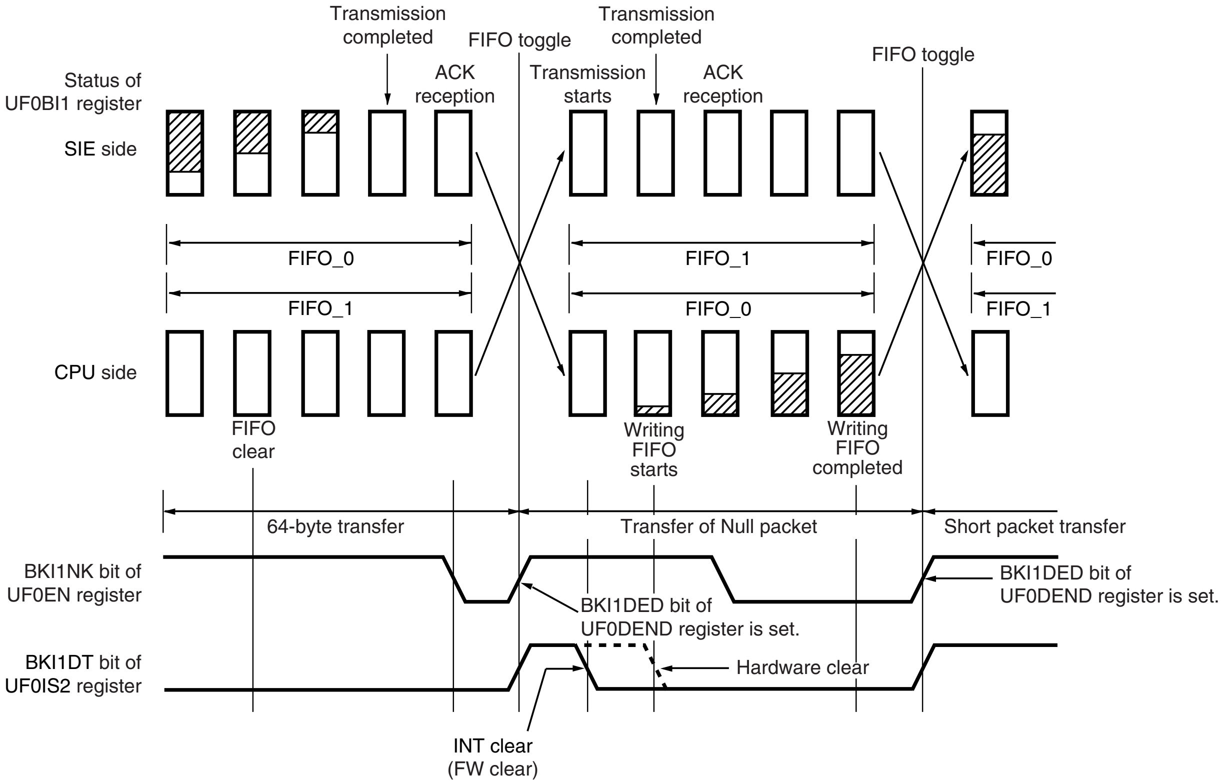


Figure 18-9. Operation of UF0BI1 Register (3/3)

(c) Operation example 3



(10) UF0 bulk-in 2 register (UF0BI2)

The UF0BI2 register is a 64-byte × 2 FIFO that stores data for Endpoint3. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when no data is in the FIFO on the SIE side (counter value = 0) and when the FIFO on the CPU side is correctly written (FIFO full or BKI2DED bit = 1).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint3 only when the BKI2NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). The address at which data is to be written or read is managed by the hardware. Therefore, FW can transmit data to the host only by writing the data to the UF0BI2 register sequentially. A short packet is transmitted when data is written to the UF0BI2 register and the BKI2DED bit of the UF0DEND register is set to 1 (BKIN2 bit of UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0BI2 register is cleared and the BKI2DED bit of the UF0DEND register is set to 1 (BKIN2 bit of the UF0EPS0 register = 1 (data exists)). When the data is transmitted correctly, a FIFO toggle operation occurs. The BKI2DT bit of the UF0IS2 register is set to 1, and an interrupt request is generated for the CPU.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BI2	BKI27	BKI26	BKI25	BKI24	BKI23	BKI22	BKI21	BKI20	00400112H	Undefined

Bit position	Bit name	Function
7 to 0	BKI27 to BKI20	These bits store data for Endpoint3.

The operation of the UF0BI2 register is illustrated below.

Figure 18-10. Operation of UF0BI2 Register (1/3)

(a) Operation example 1

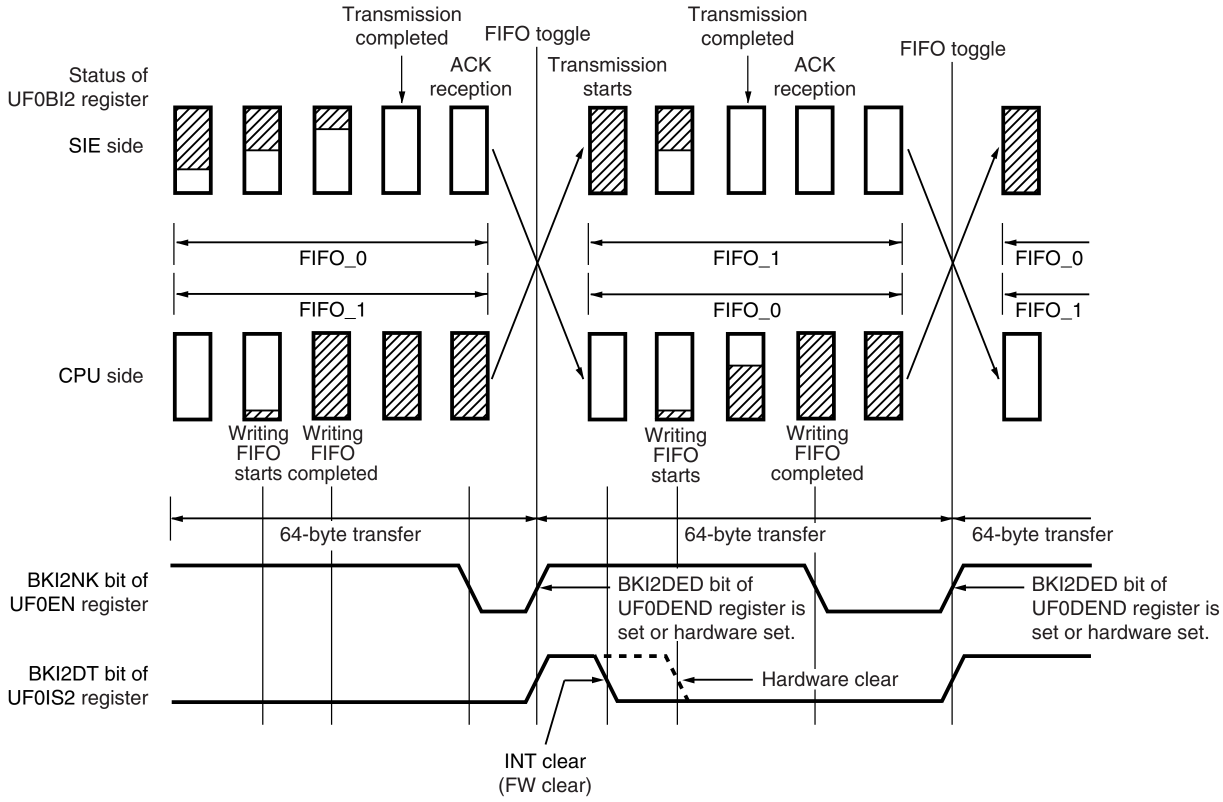


Figure 18-10. Operation of UF0BI2 Register (2/3)

(b) Operation example 2

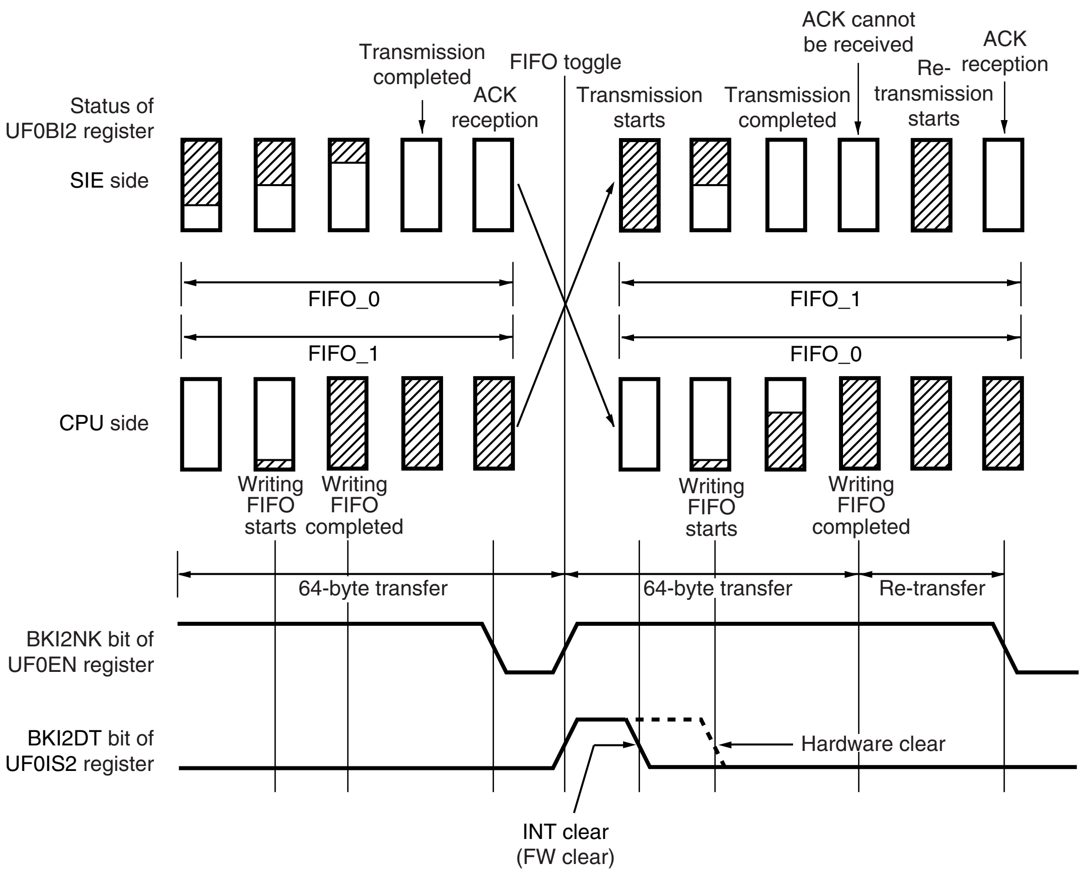
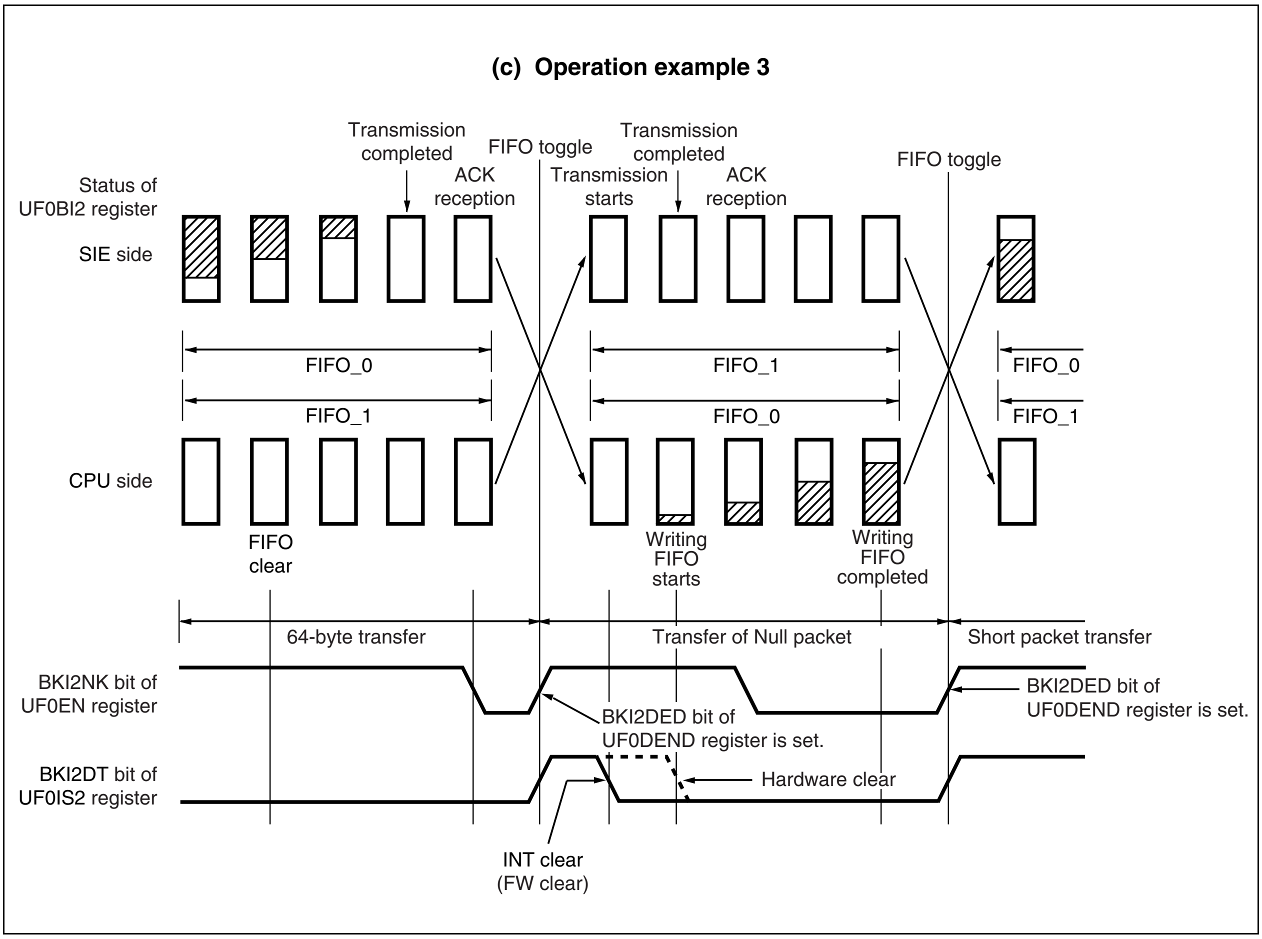


Figure 18-10. Operation of UF0BI2 Register (3/3)



(11) UF0 interrupt 1 register (UF0INT1)

The UF0INT1 register is an 8-byte FIFO that stores data for Endpoint7 (to be passed to SIE).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

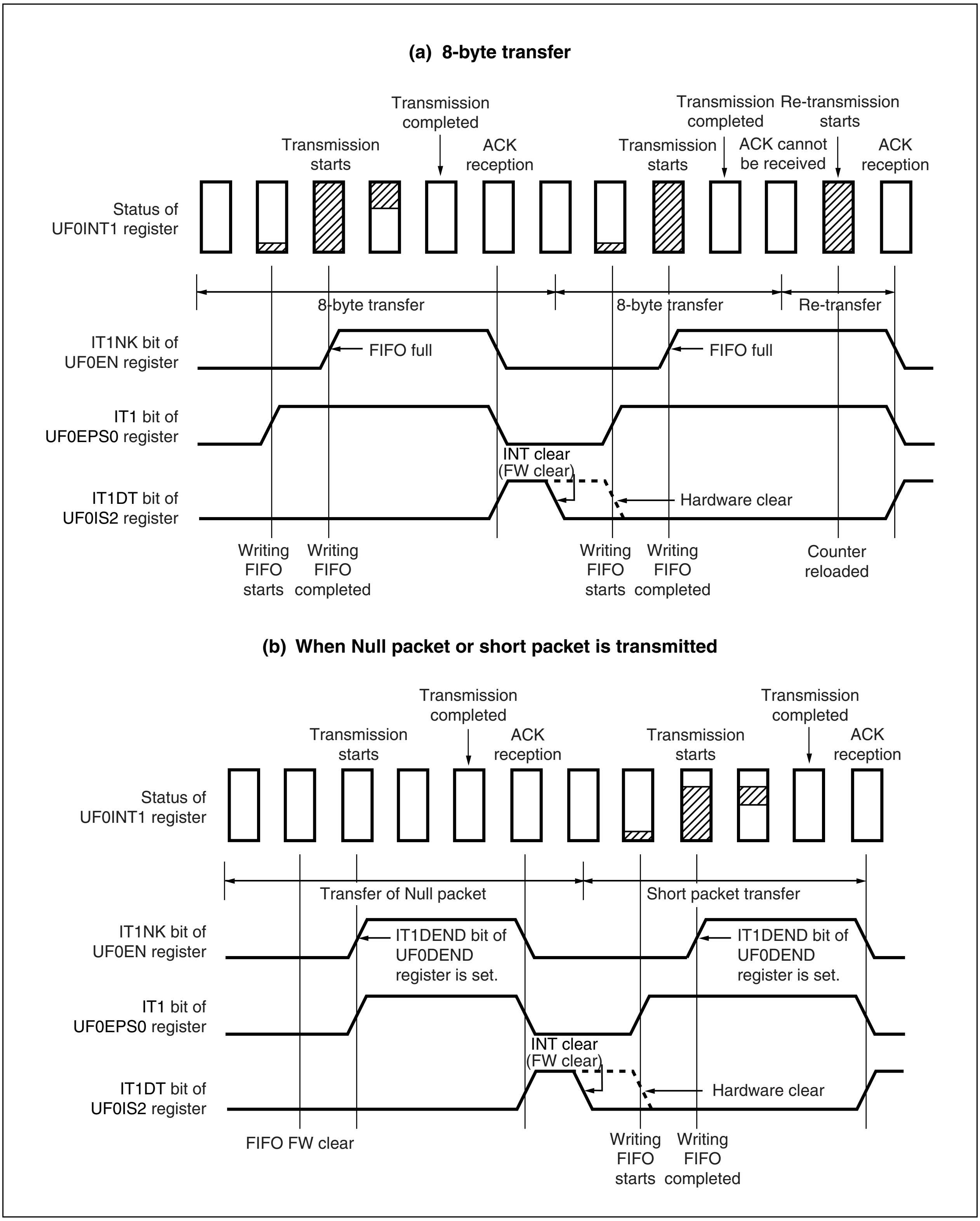
The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint7 only when the IT1NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). When the data is transmitted and the host correctly receives it, the IT1NK bit of the UF0EN register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the UF0INT1 register and the IT1DEND bit of the UF0DEND register is set to 1 (IT1 bit of the UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0INT1 register is cleared and the IT1DEND bit of the UF0DEND register is set to 1 (IT1 bit of the UF0EPS0 register = 1 (data exists)).

	7	6	5	4	3	2	1	0	Address	After reset
UF0INT1	IT17	IT16	IT15	IT14	IT13	IT12	IT11	IT10	00400114H	Undefined

Bit position	Bit name	Function
7 to 0	IT17 to IT10	These bits store data for Endpoint7.

The operation of the UF0INT1 register is illustrated below.

Figure 18-11. Operation of UF0INT1 Register



18.6.5 EPC request data registers

(1) UF0 device status register L (UF0DSTL)

This register stores the value that is to be returned in response to the GET_STATUS Device request.

This register can be read or written in 8-bit units.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Device request.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DSTL	0	0	0	0	0	0	RMWK	SFPW	00400144H	00H

Bit position	Bit name	Function
1	RMWK	This bit specifies whether the remote wakeup function of the device is used. 1: Enabled 0: Disabled If the device supports a remote wakeup function, this bit is set to 1 by hardware when the SET_FEATURE Device request has been received, and is cleared to 0 by hardware when the CLEAR_FEATURE Device request has been received. If the device does not support a remote wakeup function, make sure that the SET_FEATURE Device request is not issued from the host.
0	SFPW	This bit indicates whether the device is self-powered or bus-powered. 1: Self-powered 0: Bus-powered

(2) UF0 EP0 status register L (UF0E0SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint0 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in USBF, the E0HALT bit is set to 1 by FW. A write access to this register is ignored while a USB-side access to Endpoint0 is being received.

When the E0HALT bit is set to 1 by FW, it is not reflected until the next SETUP token is received if the control transfer immediately before is for the SET_FEATURE Endpoint0, CLEAR_FEATURE Endpoint0, GET_STATUS Endpoint0 request, or an FW-processed request.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint0 request. If Endpoint0 has stalled, the UF0E0W and UF0E0R registers are cleared, and the EP0NKW and EP0NKR bits of the UF0E0N register are cleared to 0.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0SL	0	0	0	0	0	0	0	E0HALT	0040014CH	00H

Bit position	Bit name	Function
0	E0HALT	This bit indicates the status of Endpoint0. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint0 request has been received, and cleared to 0 by hardware when the CLEAR_FEATURE Endpoint0 request has been received. DATA PID is initialized to DATA0.

(3) UF0 EP1 status register L (UF0E1SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint1 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint1, the E1HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint1 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint1 request. If Endpoint1 has stalled, the UF0BI1 register is cleared and the BKI1NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint1, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E1SL	0	0	0	0	0	0	0	E1HALT	00400150H	00H

Bit position	Bit name	Function
0	E1HALT	This bit indicates the status of Endpoint1. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint1 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint1 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint1 is linked has correctly been received. DATA PID is initialized to DATA0.

(4) UF0 EP2 status register L (UF0E2SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint2 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint2, the E2HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint2 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint2 request. If Endpoint2 has stalled, the UF0BO1 register is cleared and the BKO1NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint2, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E2SL	0	0	0	0	0	0	0	E2HALT	00400154H	00H

Bit position	Bit name	Function
0	E2HALT	This bit indicates the status of Endpoint2. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint2 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint2 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint2 is linked has correctly been received. DATA PID is initialized to DATA0.

(5) UF0 EP3 status register L (UF0E3SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint3 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint3, the E3HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint3 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint3 request. If Endpoint3 has stalled, the UF0BI2 register is cleared and the BKI2NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint3, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E3SL	0	0	0	0	0	0	0	E3HALT	00400158H	00H

Bit position	Bit name	Function
0	E3HALT	This bit indicates the status of Endpoint3. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint3 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint3 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint3 is linked has correctly been received. DATA PID is initialized to DATA0.

(6) UF0 EP4 status register L (UF0E4SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint4 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint4, the E4HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint4 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint4 request. If Endpoint4 has stalled, the UF0BO2 register is cleared and the BKO2NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint4, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E4SL	0	0	0	0	0	0	0	E4HALT	0040015CH	00H

Bit position	Bit name	Function
0	E4HALT	This bit indicates the status of Endpoint4. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint4 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint4 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint4 is linked has correctly been received. DATA PID is initialized to DATA0.

(7) UF0 EP7 status register L (UF0E7SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint7 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint7, the E7HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint7 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint7 request. If Endpoint7 has stalled, the UF0INT1 register is cleared and the IT1NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint7, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E7SL	0	0	0	0	0	0	0	E7HALT	00400168H	00H

Bit position	Bit name	Function
0	E7HALT	This bit indicates the status of Endpoint7. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint7 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint7 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint7 is linked has correctly been received. DATA PID is initialized to DATA0.

(8) UF0 address register (UF0ADRS)

This register stores the device address.

This register is read-only, in 8-bit units.

The device address sent by the SET_ADDRESS request is analyzed and the resultant value is automatically written to this register. If the SET_ADDRESS request is processed by FW, the value of this register is reflected as the device address when the SUCCESS signal is received in the status stage.

Caution Do not execute a write access to this register. If written, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ADRS	0	ADRS6	ADRS5	ADRS4	ADRS3	ADRS2	ADRS1	ADRS0	00400180H	00H

Bit position	Bit name	Function
6 to 0	ADRS6 to ADRS0	These bits hold the device address of SIE.

(9) UF0 configuration register (UF0CNF)

This register stores the value that is to be returned in response to the GET_CONFIGURATION request.

This register is read-only, in 8-bit units.

When the SET_CONFIGURATION request is received, its wValue is automatically written to this register.

When a change of the value of this register from 00H to other than 00H is detected, the CONF bit of the UF0MODS register is set to 1. If the SET_CONFIGURATION request is processed by FW, the status of this register is immediately reflected on the UF0MODS register as soon as data has been written to this register (CONF bit = 1 before completion of the status stage).

Caution Do not execute a write access to this register. If written, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CNF	0	0	0	0	0	0	CONF1	CONF0	00400182H	00H

Bit position	Bit name	Function
1, 0	CONF1, CONF0	These bits hold the data to be returned in response to the GET_CONFIGURATION request.

(10) UF0 interface 0 register (UF0IF0)

This register stores the value that is to be returned in response to the GET_INTERFACE wIndex = 0 request.

This register is read-only, in 8-bit units.

When the SET_INTERFACE request is received, its wValue is automatically written to this register.

If the SET_INTERFACE request is processed by FW, wIndex and wValue are decoded, and the setting of endpoint is automatically changed. At this time, the status bit of the target endpoint and DPID are automatically cleared to 0, depending on the setting. The FIFO is not cleared automatically.

Caution Do not execute a write access to this register. If written, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IF0	0	0	0	0	0	IF02	IF01	IF00	00400184H	00H

Bit position	Bit name	Function
2 to 0	IF02 to IF00	These bits hold the data to be returned in response to GET_INTERFACE wIndex = 0 request.

(11) UF0 interface 1 to 4 registers (UF0IF1 to UF0IF4)

These registers store the value that is to be returned in response to the GET_INTERFACE wIndex = n request (n = 1 to 4).

These registers are read-only, in 8-bit units.

When the SET_INTERFACE request is received, its wValue is automatically written to these registers.

These registers are invalidated according to the setting of the UF0AIFN and UF0AAS registers.

If the SET_INTERFACE request is processed by FW, wIndex and wValue are decoded, and the setting of endpoint is automatically changed. At this time, the status bit of the target endpoint and DPID are automatically cleared to 0, depending on the setting. The FIFO is not cleared automatically.

Caution Do not execute a write access to this register. If written, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IF1	0	0	0	0	0	IF12	IF11	IF10	00400186H	00H
UF0IF2	0	0	0	0	0	IF22	IF21	IF20	00400188H	00H
UF0IF3	0	0	0	0	0	IF32	IF31	IF30	0040018AH	00H
UF0IF4	0	0	0	0	0	IF42	IF41	IF40	0040018CH	00H

Bit position	Bit name	Function
2 to 0	IFn2 to IFn0	These bits hold the data to be returned in response to GET_INTERFACE wIndex = n request.

Remark n = 1 to 4

(12) UF0 descriptor length register (UF0DSCL)

This register stores the length of the value that is to be returned in response to the GET_DESCRIPTOR Configuration request. The value of this register is the number of bytes of all the descriptors set by the UF0CIEn register minus 1 (n = 0 to 255). The total descriptor length that is to be returned in response to the GET_DESCRIPTOR Configuration request is determined according to the value of this register.

This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Processing of wLength is automatically controlled. If this register is set to 00H, it means that the descriptor to be returned is 1 byte long. If the register is set to FFH, a descriptor length of 256 bytes is returned. When a descriptor exceeding 256 bytes in length is used, set the CDCGDST bit of the UF0MODC register to 1 and process the GET_DESCRIPTOR request by FW (at this time, the CDCGD bit of the UF0MODS register is also set to 1).

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DSCL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0	004001A0H	00H

Bit position	Bit name	Function
7 to 0	DPL7 to DPL0	These bits set the value of the number of bytes of all the descriptors to be returned in response to the GET_DESCRIPTOR Configuration request minus 1.

(13) UF0 device descriptor registers 0 to 17 (UF0DD0 to UF0DD17)

These registers store the value to be returned in response to the GET_DESCRIPTOR Device request.

These registers can be read or written in 8-bit units. However, data can be written to these registers only when the EP0NKA bit is set to 1.

Cautions 1. To rewrite these registers, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

2. Use the value defined by USB Specification Ver. 2.0 and the latest Class Specification as the set value.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DDn (n = 0 to 17)									See **Table 18-5**.	Undefined

Table 18-5. Mapping and Data of UF0 Device Descriptor Registers

Symbol	Address	Field Name	Contents
UF0DD0	004001A2H	bLength	Size of this descriptor
UF0DD1	004001A4H	bDescriptorType	Device descriptor type
UF0DD2	004001A6H	bcdUSB	Value below decimal point of Rev. number of USB specification
UF0DD3	004001A8H		Value above decimal point of Rev. number of USB specification
UF0DD4	004001AAH	bDeviceClass	Class code
UF0DD5	004001ACH	bDeviceSubClass	Subclass code
UF0DD6	004001AEH	bDeviceProtocol	Protocol code
UF0DD7	004001B0H	bMaxPacketSize0	Maximum packet size of Endpoint0
UF0DD8	004001B2H	idVendor	Lower value of vendor ID
UF0DD9	004001B4H		Higher value of vendor ID
UF0DD10	004001B6H	idProduct	Lower value of product ID
UF0DD11	004001B8H		Higher value of product ID
UF0DD12	004001BAH	bcdDevice	Lower value of device release number
UF0DD13	004001BCH		Higher value of device release number
UF0DD14	004001BEH	iManufacturer	Index of string descriptor describing manufacturer
UF0DD15	004001C0H	iProduct	Index of string descriptor describing product
UF0DD16	004001C2H	ISerialNumber	Index of string descriptor describing device serial number
UF0DD17	004001C4H	BNumConfigurations	Number of settable configurations

(14) UF0 configuration/interface/endpoint descriptor registers 0 to 255 (UF0CIE0 to UF0CIE255)

These registers store the value to be returned in response to the GET_DESCRIPTOR Configuration request.

These registers can be read or written in 8-bit units. However, data can be written to these registers only when the EP0NKA bit is set to 1.

Descriptor information of up to 256 bytes can be stored in these registers. Store each descriptor in the order of Configuration, Interface, and Endpoint (see **Table 18-6**). If there are two or more Interfaces, repeatedly store the data following the Interface descriptor.

Table 18-6. Mapping of UF0CIEn Register

Address	Descriptor Stored
004001C6H	Configuration descriptor (9 bytes)
004001D8H	Interface descriptor (9 bytes)
004001EAH	Endpoint1 descriptor (7 bytes)
004001F8H	Endpoint2 descriptor (7 bytes)
00400206H	Endpoint3 descriptor (7 bytes)
:	:
004002xxH	Interface descriptor (9 bytes)
004002xxH+9	Endpoint1 descriptor (7 bytes)
004002xxH+16	Endpoint2 descriptor (7 bytes)
004002xxH+23	Endpoint3 descriptor (7 bytes)
:	:

The range of the valid data that can be set to these registers varies according to the setting of the UF0DSCL register. In addition to the descriptors listed in Table 18-7, descriptors peculiar to classes and vendors can also be stored.

If all the values are fixed, they can be stored in ROM.

Cautions 1. To rewrite these registers, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

2. Use the value defined by USB Specification Ver. 2.0 and the latest Class Specification as the set value.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CIEn (n = 0 to 255)									004001C6H to 004003C4H	Undefined

Table 18-7. Data of UF0CIEn Register

(a) Configuration descriptor (9 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	wTotalLength	Lower value of the total number of bytes of Configuration, all Interface, and all Endpoint descriptors
3		Higher value of the total number of bytes of Configuration, all Interface, and all Endpoint descriptors
4	bNumInterface	Number of Interfaces
5	bConfigurationValue	Value to select this Configuration
6	iConfiguration	Index of string descriptor describing this Configuration
7	bmAttributes	Features of this Configuration (self-powered, without remote wakeup)
8	MaxPower	Maximum power consumption of this Configuration (unit: mA)[Note]

Note Shown in 2 mA units. (example: 50 = 100 mA)

(b) Interface descriptor (9 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	bInterfaceNumber	Value of this Interface
3	bAlternateSetting	Value to select alternative setting of Interface
4	bNumEndpoints	Number of usable Endpoints
5	bInterfaceClass	Class code
6	bInterfaceSubClass	Subclass code
7	bInterfaceProtocol	Protocol code
8	Interface	Index of string descriptor describing this Interface

(c) Endpoint descriptor (7 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	bEndpointAddress	Address/transfer direction of this Endpoint
3	bmAttributes	Transfer type
4	wMaxPaketSize	Lower value of maximum number of transfer data
5		Higher value of maximum number of transfer data
6	bInterval	Transfer interval

18.6.6 Bridge register

(1) Bridge interrupt control register (BRGINTT)

The BRGINTT register controls the status of interrupts generated from EPC to the bridge circuit.

The BRGINTT register can be read or written in 16-bit units.

Be sure to clear bits 3 to 15 to “0”.

After reset: 0000H R/W Address: 00400400H

	15	14	13	12	11	10	9	8
BRGINTT	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	0	0	0	0	0	EPCINT2B	EPCINT1B	EPCINT0B

Bit position	Bit name	Function
2	EPCINT2B	Shows the status of the interrupt signal “EPC_INT2B” from EPC. Clears the request of EPC register. 0: Interrupt not issued 1: Interrupt issued
1	EPCINT1B	Shows the status of the interrupt signal “EPC_INT1B” from EPC. Clears the request of EPC register. 0: Interrupt not issued 1: Interrupt issued
0	EPCINT0B	Shows the status of the interrupt signal ”EPC_INT0B” from EPC. Clears the request of EPC register. 0: Interrupt not issued 1: Interrupt issued

(2) Bridge interrupt enable register (BRGINTE)

The BRGINTE register controls whether the interrupt generated in the bridge circuit is enabled or disabled.

The BRGINTE register can be read or written in 16-bit units.

Be sure to clear bits 3 to 15 to “0”.

After reset: 0000H R/W Address: 00400402H

	15	14	13	12	11	10	9	8
BRGINTE	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	0	0	0	0	0	EPC INT2BEN	EPC INT1BEN	EPC INT0BEN

Bit position	Bit name	Function
2	EPCINT2BEN	Enables or disables interrupt occurrence when EPCINT2BEN bit is set. 0: Interrupt disabled 1: Interrupt enabled
1	EPCINT1BEN	Setting the interrupt occur enable or disable when EPCINT1BEN bit is setting. 0: Interrupt disabled 1: Interrupt enabled
0	EPCINT0BEN	Setting the interrupt occur enable or disable when EPCINT0BEN bit is setting. 0: Interrupt disabled 1: Interrupt enabled

(3) EPC macro control register (EPCCLT)

The EPCCLT register controls the reset generator to the EPC macro.

The EPCCLT register can be read or written in 16-bit units.

After reset: 0000H R/W Address: 00400404H

	15	14	13	12	11	10	9	8
EPCCLT	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	EPCRST

Bit position	Bit name	Function
0	EPCRST	Sets the reset for EPC. 0: Reset released 1: Reset issued

(4) CPU I/F bus control register (CPUBCTL)

The CPUBCTL register controls the interface between bridge circuit and CPU.

The CPUBCTL register can be read or written in 16-bit units.

After reset: Undefined R/W Address: 00400408H

	15	14	13	12	11	10	9	8
CPUBCTL	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	0	0	0	0	0	BULKWAIT	DATAWAIT	NOWAIT

Bit position	Bit name	Function
2	BULKWAIT	Forcibly inserts 1 wait (bulk wait) when the bulk register is accessed. 0: Do not forcibly insert the bulk wait[Note] (default value) 1: Forcibly insert the bulk wait **Note** This setting is invalid when writing: the bulk wait is always forcibly inserted.
1	DATAWAIT	Forcibly inserting the 1 wait (data wait) after the CPU bus cycle. 0: No forcibly insert the data wait (default value) 1: Forcibly insert the data wait
0	NOWAIT	Enables or disables the no wait operation of the CPU bus cycle. 0: No wait disabled[Note] (default value) 1: No wait enabled **Note** 1 wait or more is inserted.

The following flowcharts illustrate the program execution when the host is disconnected and then reconnected, and the program execution when power is supplied.

Figure 18-12. Flowchart of Program When Host Is Disconnected and Then Reconnected

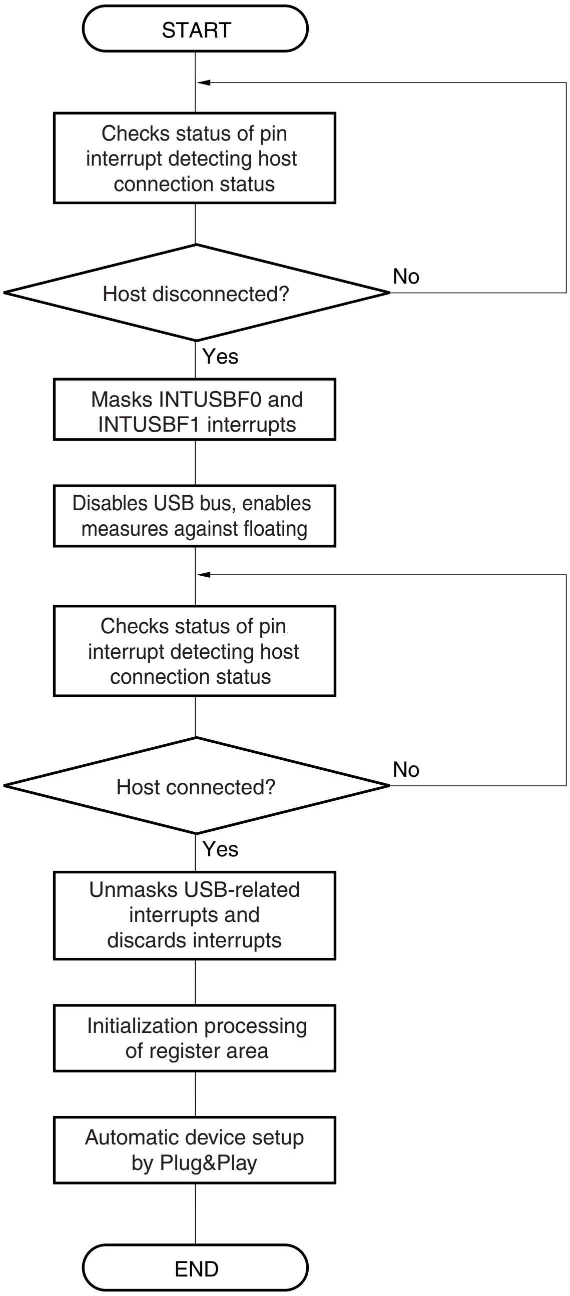
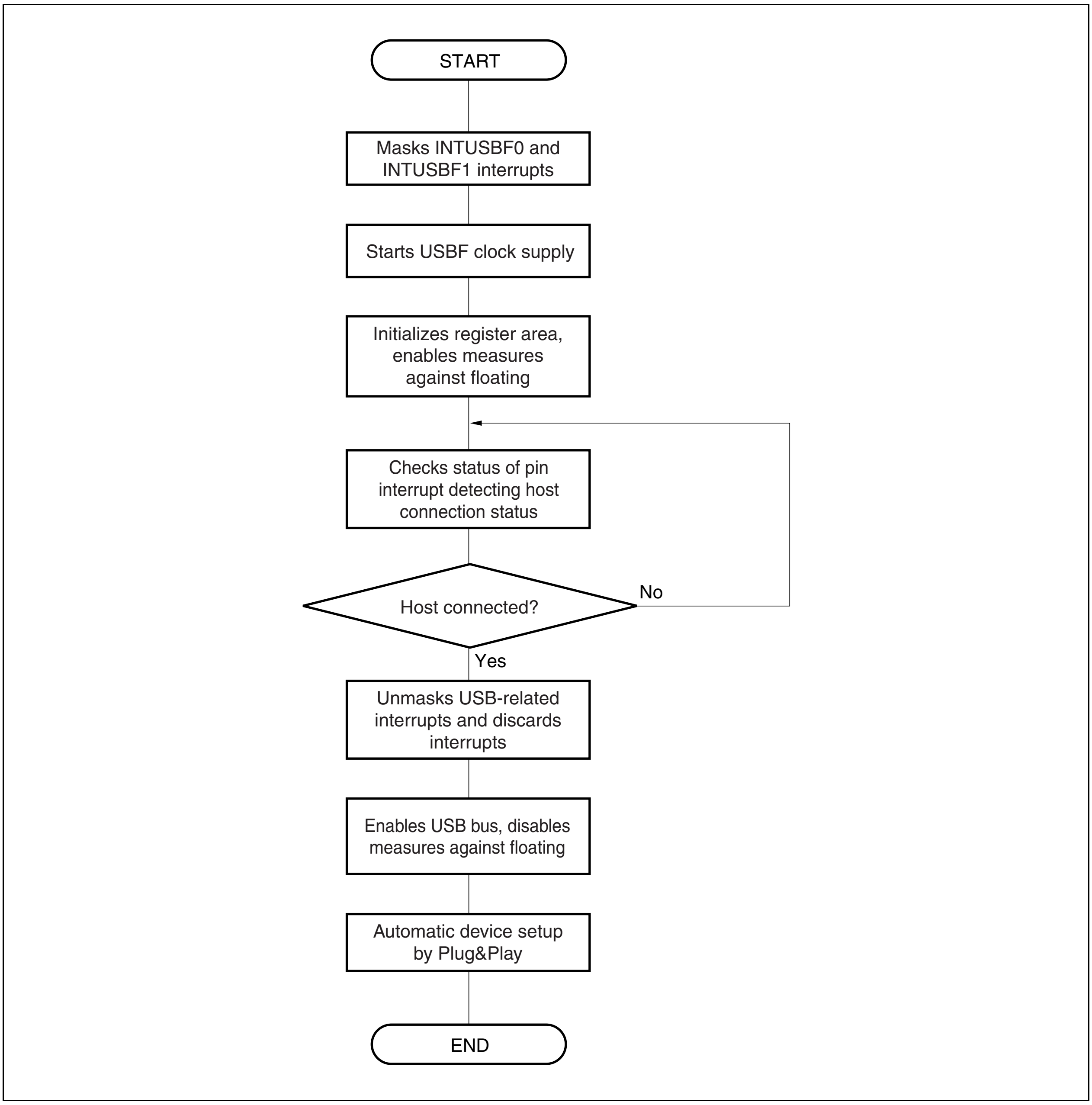

START
Checks status of pin interrupt detecting host connection status
Host disconnected?
No
Yes
Masks INTUSBF0 and INTUSBF1 interrupts
Disables USB bus, enables measures against floating
Checks status of pin interrupt detecting host connection status
Host connected?
No
Yes
Unmasks USB-related interrupts and discards interrupts
Initialization processing of register area
Automatic device setup by Plug&Play
END

Figure 18-13. Flowchart of Program When Power Is Supplied


START
Masks INTUSBF0 and INTUSBF1 interrupts
Starts USBF clock supply
Initializes register area, enables measures against floating
Checks status of pin interrupt detecting host connection status
Host connected?
No
Yes
Unmasks USB-related interrupts and discards interrupts
Enables USB bus, disables measures against floating
Automatic device setup by Plug&Play
END

18.7 STALL Handshake or No Handshake

Errors of USBF are defined to be handled as follows.

Transfer Type	Transaction	Target Packet	Error Type	Function Response	Processing
Control transfer/ bulk transfer/ interrupt transfer	IN/OUT/SETUP	Token	Endpoint not supported	No response	None
			Endpoint transfer direction mismatch	No response	None
			CRC error	No response	None
			Bit stuffing error	No response	None
Control transfer/ bulk transfer	OUT/SETUP	Data	Timeout	No response	None
			PID check error	No response	None
			Unsupported PID (other than Data PID)	No response	None
			CRC error	No response	Discard received data
			Bit stuffing error	No response	Discard received data
	OUT	Data	Data PID mismatch	ACK	Discard received data
Control transfer (SETUP stage)	SETUP	Data	Overrun	No response	Discard received data
Control transfer (data stage)	OUT	Data	Overrun	No response[Note 1]	Set SNDSTL bit of UF0SDS register to 1 and discard received data
Control transfer (status stage)	OUT	Data	Overrun	ACK or no response[Note 2]	Set SNDSTL bit of UF0SDS register to 1 and discard received data
Bulk transfer	OUT	Data	Overrun	No response[Note 1]	Set EnHALT bit of UF0EnSL register (n = 0 to 4, 7) to 1
Control transfer/ bulk transfer/ interrupt transfer	IN	Handshake	PID check error	–	Hold transferred data and re-transfer data[Note 3]
			Unsupported PID (other than ACK PID)	–	Hold transferred data and re-transfer data[Note 3]
			Timeout	–	Hold transferred data and re-transfer data[Note 3]

Notes 1. A STALL is sent in response to re-transfer by the host.

2. An ACK response is made if the transfer data is less than MaxPacketSize and the data received in the status stage is discarded. If MaxPacketSize is exceeded, no response is made, the SNDSTL bit of the UF0SDS register is set to 1, and the received data is discarded.

3. If an OUT transaction indicating a change from the data stage to the status stage is received during control transfer, it is not handled as an error and it is assumed that reception has been correctly completed.

Cautions 1. It is judged by the Alternative Setting number currently set whether the target Endpoint is valid or invalid.

2. For the response to the request included in control transfer to/from Endpoint0, see 18.5 Requests.

18.8 Register Values in Specific Status

Table 18-8. Register Values in Specific Status (1/2)

Register Name	After CPU Reset ($\overline{\text{RESET}}$)	After Bus Reset
UF0E0N register	00H	Value is held.
UF0E0NA register	00H	Value is held.
UF0EN register	00H	Value is held.
UF0ENM register	00H	Value is held.
UF0SDS register	00H	Value is held.
UF0CLR register	00H	Value is held.
UF0SET register	00H	Value is held.
UF0EPS0 register	00H	Value is held.
UF0EPS1 register	00H	Value is held.
UF0EPS2 register	00H	Value is held.
UF0IS0 register	00H	Value is held.
UF0IS1 register	00H	Value is held.
UF0IS2 register	00H	Value is held.
UF0IS3 register	00H	Value is held.
UF0IS4 register	00H	Value is held.
UF0IM0 register	00H	Value is held.
UF0IM1 register	00H	Value is held.
UF0IM2 register	00H	Value is held.
UF0IM3 register	00H	Value is held.
UF0IM4 register	00H	Value is held.
UF0IC0 register	FFH	Value is held.
UF0IC1 register	FFH	Value is held.
UF0IC2 register	FFH	Value is held.
UF0IC3 register	FFH	Value is held.
UF0IC4 register	FFH	Value is held.
UF0FIC0 register	00H	Value is held.
UF0FIC1 register	00H	Value is held.
UF0DEND register	00H	Value is held.
UF0GPR register	00H	Value is held.
UF0MODC register	00H	Value is held.
UF0MODS register	00H	Bit 2 (CONF): Cleared (0), Other bits: Value is held.
UF0AIFN register	00H	Value is held.
UF0AAS register	00H	Value is held.
UF0ASS register	00H	00H
UF0E1IM register	00H	Value is held.
UF0E2IM register	00H	Value is held.

Table 18-8. Register Values in Specific Status (2/2)

Register Name	After CPU Reset ($\overline{\text{RESET}}$)	After Bus Reset
UF0E3IM register	00H	Value is held.
UF0E4IM register	00H	Value is held.
UF0E7IM register	00H	Value is held.
UF0E0R register	Undefined[Note 1]	Value is held.
UF0E0L register	00H	Value is held.
UF0E0ST register	00H	00H
UF0E0W register	Undefined[Note 1]	Value is held.
UF0BO1 register	Undefined[Note 1]	Value is held.
UF0BO1L register	00H	Value is held.
UF0BO2 register	Undefined[Note 1]	Value is held.
UF0BO2L register	00H	Value is held.
UF0BI1 register	Undefined[Note 1]	Value is held.
UF0BI2 register	Undefined[Note 1]	Value is held.
UF0INT1 register	Undefined	Value is held.
UF0DSTL register	00H	00H
UF0E0SL register	00H	00H
UF0E1SL register	00H	00H
UF0E2SL register	00H	00H
UF0E3SL register	00H	00H
UF0E4SL register	00H	00H
UF0E7SL register	00H	00H
UF0ADRS register	00H	00H
UF0CNF register	00H	00H
UF0IF0 register	00H	00H
UF0IF1 register	00H	00H
UF0IF2 register	00H	00H
UF0IF3 register	00H	00H
UF0IF4 register	00H	00H
UF0DSCL register	00H	Value is held.
UF0DDn register (n = 0 to 17)	**Note 2**	**Note 2**
UF0CIEn register (n = 0 to 255)	**Note 2**	**Note 2**

Notes 1. This register can be cleared to 0 by the $\overline{\text{RESET}}$ signal because its write pointer, counter, and read pointer are cleared to 0 when the $\overline{\text{RESET}}$ signal becomes active, in the same manner as clearing by the UF0FICn register, as the register is controlled by FIFO.

2. This register cannot be cleared to 0. Because data can be written to it by FW, however, any value can be written to the register (before doing so, however, be sure to set the EP0NKA bit of the UF0E0NA register to 1).

18.9 FW Processing

The following FW processing is performed.

- Setting processing on device side for the SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE requests during enumeration processing
- Analysis and processing of XXXXStandard, XXXXClass, and XXXXVendor requests not subject to automatic processing
- Reading data following bulk-transferred OUT token from receive buffer
- Writing data to be returned in response to bulk-transferred IN token
- Writing data to be returned in response to interrupt-transferred token

The following table lists the requests supported by FW.

Table 18-9. FW-Supported Standard Requests

Request	Reception Side	Processing/ Frequency	Explanation
CLEAR_FEATURE	Interface	Automatic STALL response	It is considered that this request does not come to Interface because there is no function selector value, though it is reserved for bmRequestType. When this request is received, the hardware makes an automatic STALL response.
SET_FEATURE	Interface	Automatic STALL response	It is considered that this request does not come to Interface because there is no function selector value, though it is reserved for bmRequestType. When this request is received, the hardware makes an automatic STALL response.
GET_DESCRIPTOR	String	FW	Returns the string descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and writes the data to be returned to the host, to the UF0E0W register.
SET_DESCRIPTOR	Device	FW	Rewrites the device descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and the writes the data for the next control transfer (OUT) to the UF0DDn register (n = 0 to 17).
SET_DESCRIPTOR	Configuration	FW	Rewrites the configuration descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and the writes the data for the next control transfer (OUT) to the UF0CIEn register (n = 0 to 255).
SET_DESCRIPTOR	String	FW	Rewrites the string descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and loads the data for the next control transfer (OUT).
Other	NA	FW	When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and performs the necessary processing.

18.9.1 Initialization processing

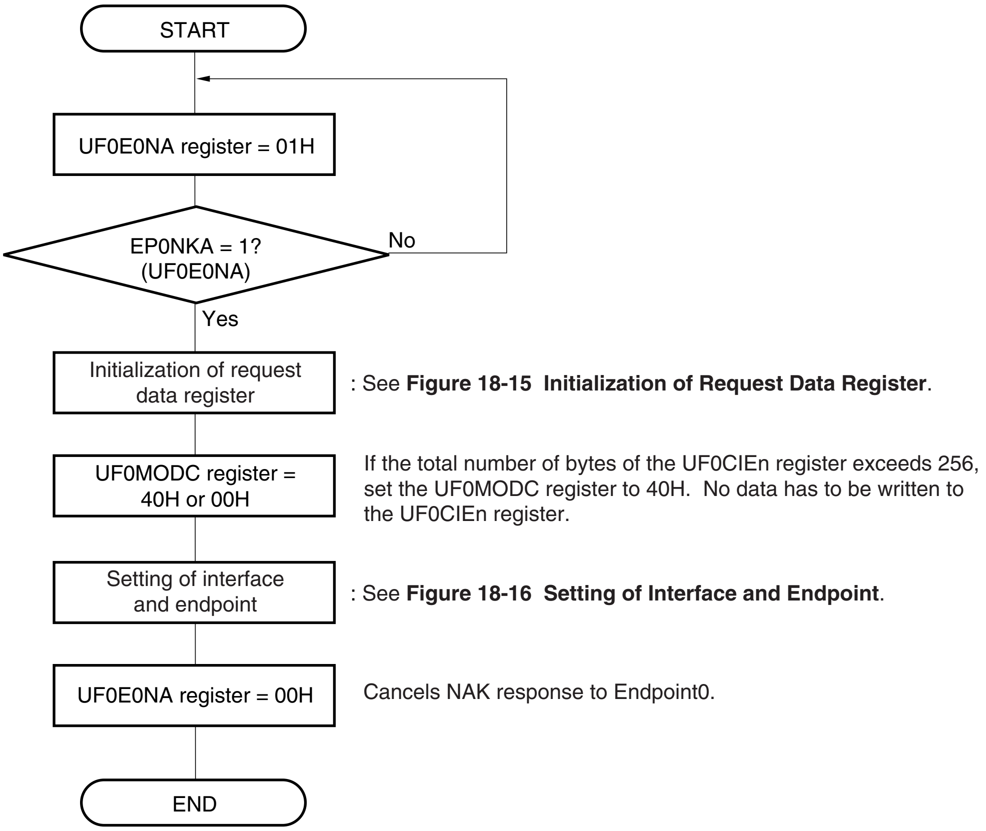
Initialization processing is executed in the following two ways.

- Initialization of request data register
- Setting of interrupt

When a request data register is initialized, data for the GET_XXXX request to which a value is to be automatically returned is written and an endpoint is allocated to an interface. In the interrupt settings, the interrupt sources that do not have to be checked can be masked by using the UF0IMn register (n = 0 to 4).

The following flowcharts illustrate the above processing.

Figure 18-14. Initializing Request Data Register


START
UF0E0NA register = 01H
EP0NKA = 1?
(UF0E0NA)
No
Yes
Initialization of request data register
UF0MODC register = 40H or 00H
Setting of interface and endpoint
UF0E0NA register = 00H
END

: See **Figure 18-15 Initialization of Request Data Register**.

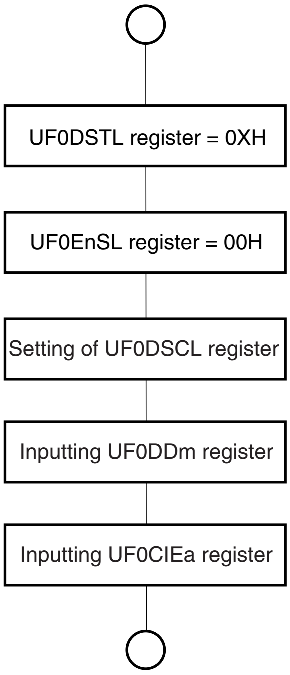
If the total number of bytes of the UF0CIEn register exceeds 256, set the UF0MODC register to 40H. No data has to be written to the UF0CIEn register.

: See **Figure 18-16 Setting of Interface and Endpoint**.

Cancels NAK response to Endpoint0.

Remark n = 0 to 255

Figure 18-15. Initialization Settings of Request Data Register


UF0DSTL register = 0XH
UF0EnSL register = 00H
Setting of UF0DSCL register
Inputting UF0DDm register
Inputting UF0CIEa register

The value of 0XH depends on the power supply method.
- SFPW = 1: Self-powered
- SFPW = 0: Bus-powered

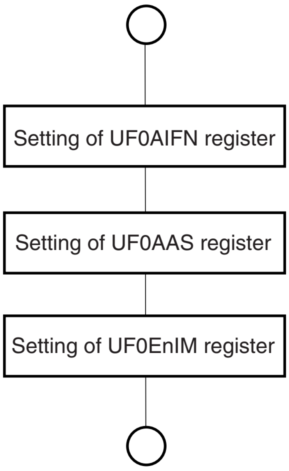
n = 0 to 4, 7. Setting is unnecessary if the target endpoint is not used.

Input the total number of bytes of the UF0CIEa register.

If the total number of bytes of the UF0CIEa register exceeds 256, set the UF0MODC register to 40H. No data has to be written to the UF0CIEa register.

Remark m = 0 to 17
a = 0 to 255

Figure 18-16. Setting of Interface and Endpoint


Setting of UF0AIFN register
Setting of UF0AAS register
Setting of UF0EnIM register

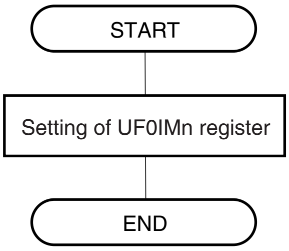
ADDIF, IFNO1, IFNO0 = 000: Interface number 0 is valid.
ADDIF, IFNO1, IFNO0 = 100: Interface numbers 0 and 1 are valid.
ADDIF, IFNO1, IFNO0 = 101: Interface numbers 0 to 2 are valid.
ADDIF, IFNO1, IFNO0 = 110: Interface numbers 0 to 3 are valid.
ADDIF, IFNO1, IFNO0 = 111: Interface numbers 0 to 4 are valid.

Set Interface number(s) and a link with the 5- or 2-series Alternative Setting.

Set a link between the target Interface of endpoint n and Alternative Setting.
Set 00H if the target endpoint is not used.

Remark n = 1 to 4, 7

Figure 18-17. Setting of Interrupt



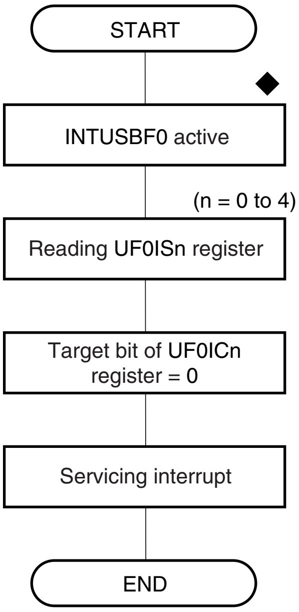
Mask the interrupt source to avoid issuance of an unnecessary interrupt request (INTUSBF0).

Remark n = 0 to 4

18.9.2 Interrupt servicing

The following flowchart illustrates how an interrupt is serviced.

Figure 18-18. Interrupt Servicing



Remark ◆: Processing by hardware

The following bits of the UF0ISn register are automatically cleared by hardware when a given condition is satisfied (n = 0 to 4).

- E0INDT, E0ODT, SUCES, STG, and CPUDEC bits of UF0IS1 register
- BKI2DT, BKI1DT, and IT1DT bits of UF0IS2 register
- BKO2FL, BKO2DT, BKO1FL, and BKO1DT bits of UF0IS3 register

Because clearing an interrupt source by the UF0ICn register is given a lower priority than setting an interrupt source by hardware, the interrupt source may not be cleared depending on the timing (n = 0 to 4).

18.9.3 USB main processing

USB main processing involves processing USB transactions. The types of transactions to be processed are as follows.

- Fully automatically processed request for control transfer
- Automatically processed requests for control transfer
  (SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, CLEAR_FEATURE)
- CPUDEC request for control transfer
- Processing for bulk transfer (IN)
- Processing for bulk transfer (OUT)
- Processing for interrupt transfer (IN)

Processing for endpoint n involves writing or reading for data transfer. The flowchart shown below is for PIO.

(1) Fully automatically processed request for control transfer

Because the fully automatically processed request for control transfer is executed by hardware, it cannot be referenced by FW. Therefore, FW does not have to perform any special processing for this request.

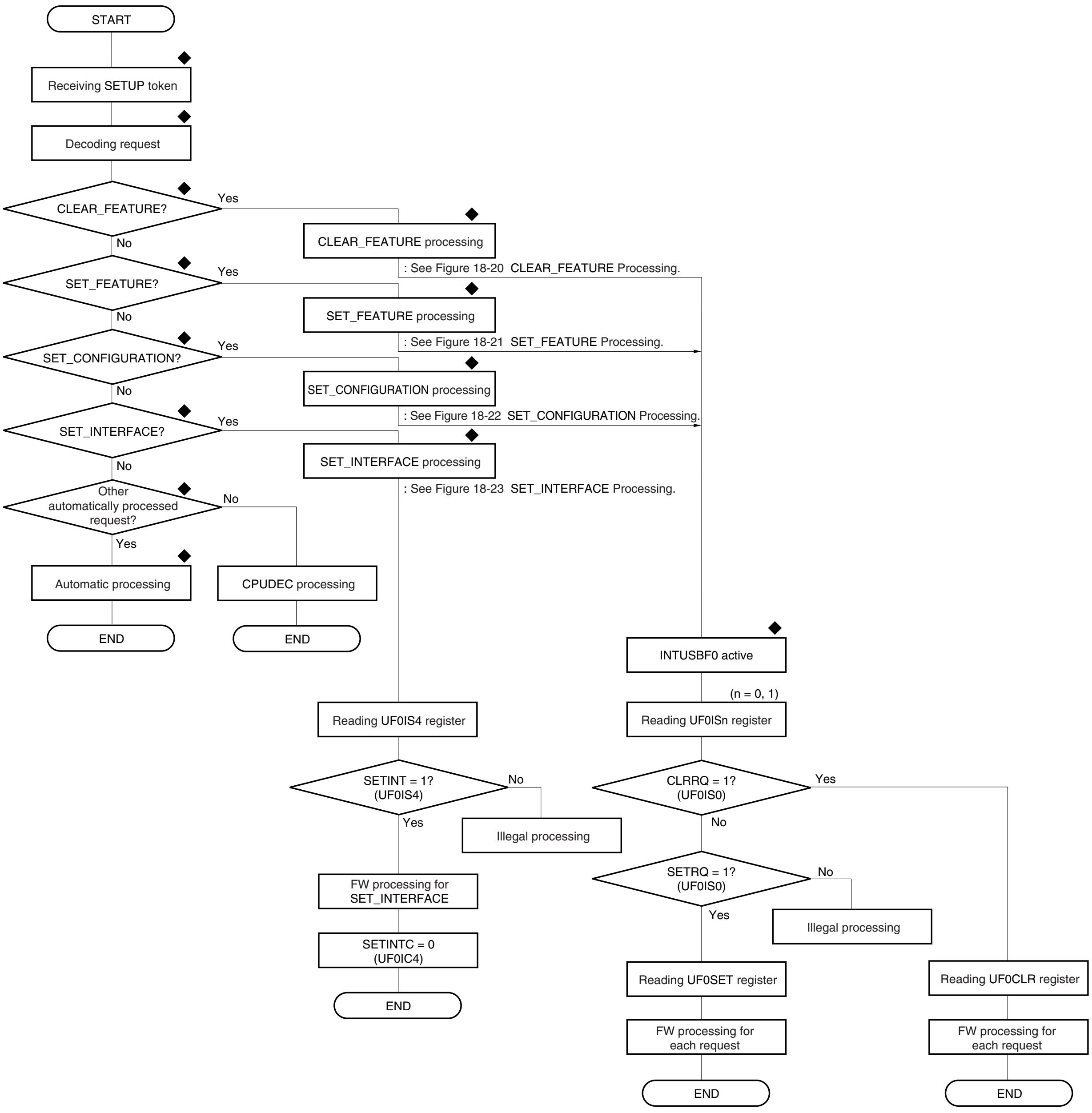
(2) Automatically processed requests for control transfer

(SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, CLEAR_FEATURE)

Processing to write a register for automatically processed requests for control transfer, such as SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE requests, is automatically executed by hardware, but an interrupt request is issued for recognition on the device side. This processing may be ignored if there is no special processing to be executed.

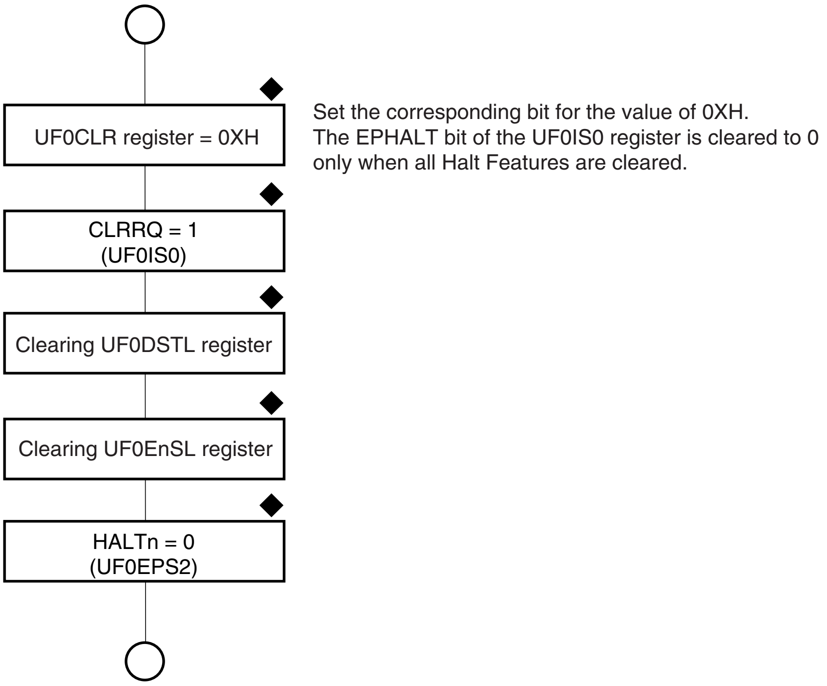
The flowcharts are shown below.

Figure 18-19. Automatically Processed Requests for Control Transfer



Remark ♦: Processing by hardware

Figure 18-20. CLEAR_FEATURE Processing

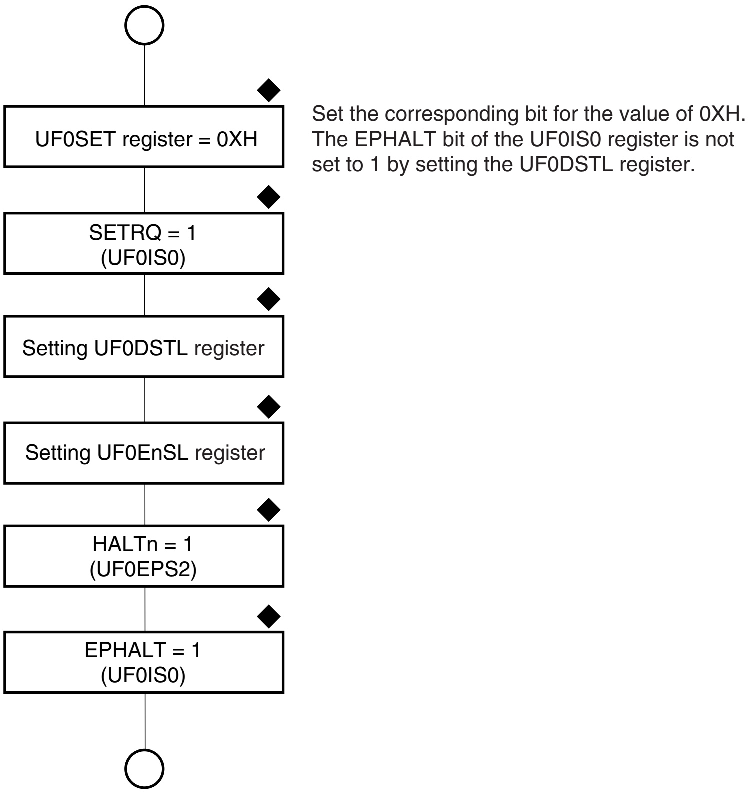


Set the corresponding bit for the value of 0XH.
The EPHALT bit of the UF0IS0 register is cleared to 0 only when all Halt Features are cleared.

Remarks 1. n = 0 to 4, 7

2. ♦: Processing by hardware

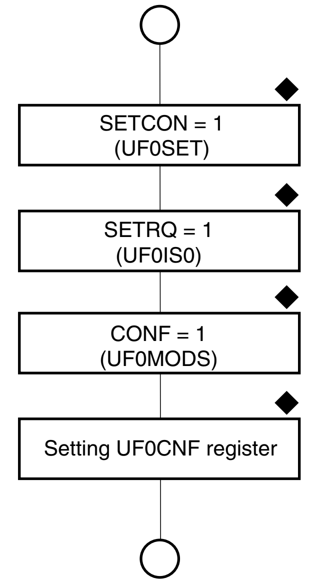
Figure 18-21. SET_FEATURE Processing


UF0SET register = 0XH
Set the corresponding bit for the value of 0XH.
The EPHALT bit of the UF0IS0 register is not
set to 1 by setting the UF0DSTL register.
SETRQ = 1
(UF0IS0)
Setting UF0DSTL register
Setting UF0EnSL register
HALTn = 1
(UF0EPS2)
EPHALT = 1
(UF0IS0)

Remarks 1. n = 0 to 4, 7

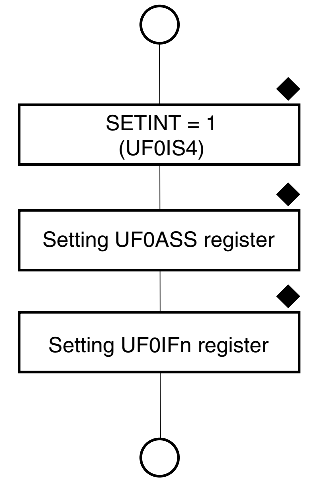
2. ♦: Processing by hardware

Figure 18-22. SET_CONFIGURATION Processing



Remark ♦: Processing by hardware

Figure 18-23. SET_INTERFACE Processing



Remarks 1. n = 0 to 4

2. ♦: Processing by hardware

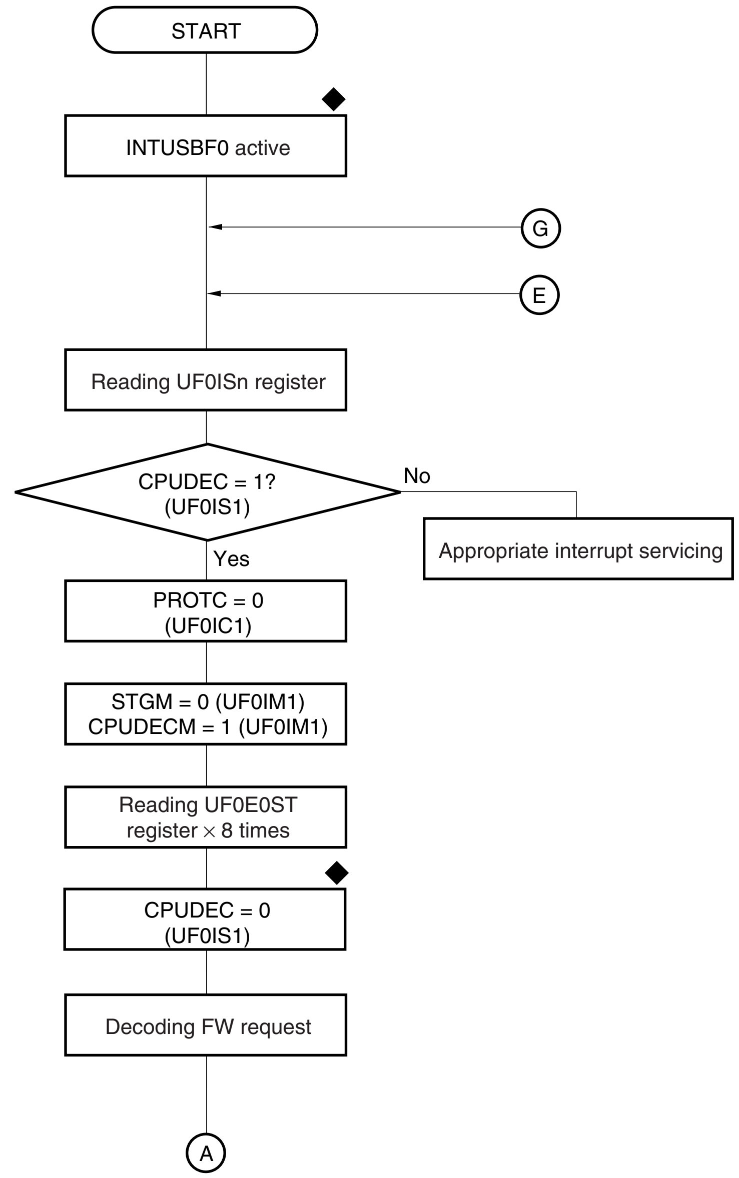
(3) CPUDEC request for control transfer

The CPUDEC request can be classified into three types of processing: control transfer (write), control transfer (read), and control transfer (without data). Control transfer (write) indicates a request that uses the OUT transaction in the data stage (e.g., SET_DESCRIPTOR), and control transfer (read) indicates a request that uses the IN transaction in the data stage (e.g., GET_DESCRIPTOR). Control transfer (without data) indicates a request that has no data stage (e.g., SET_CONFIGURATION).

The flowcharts are shown below.

Figure 18-24. CPUDEC Request for Control Transfer (1/12)

(a) Token phase (1/2)

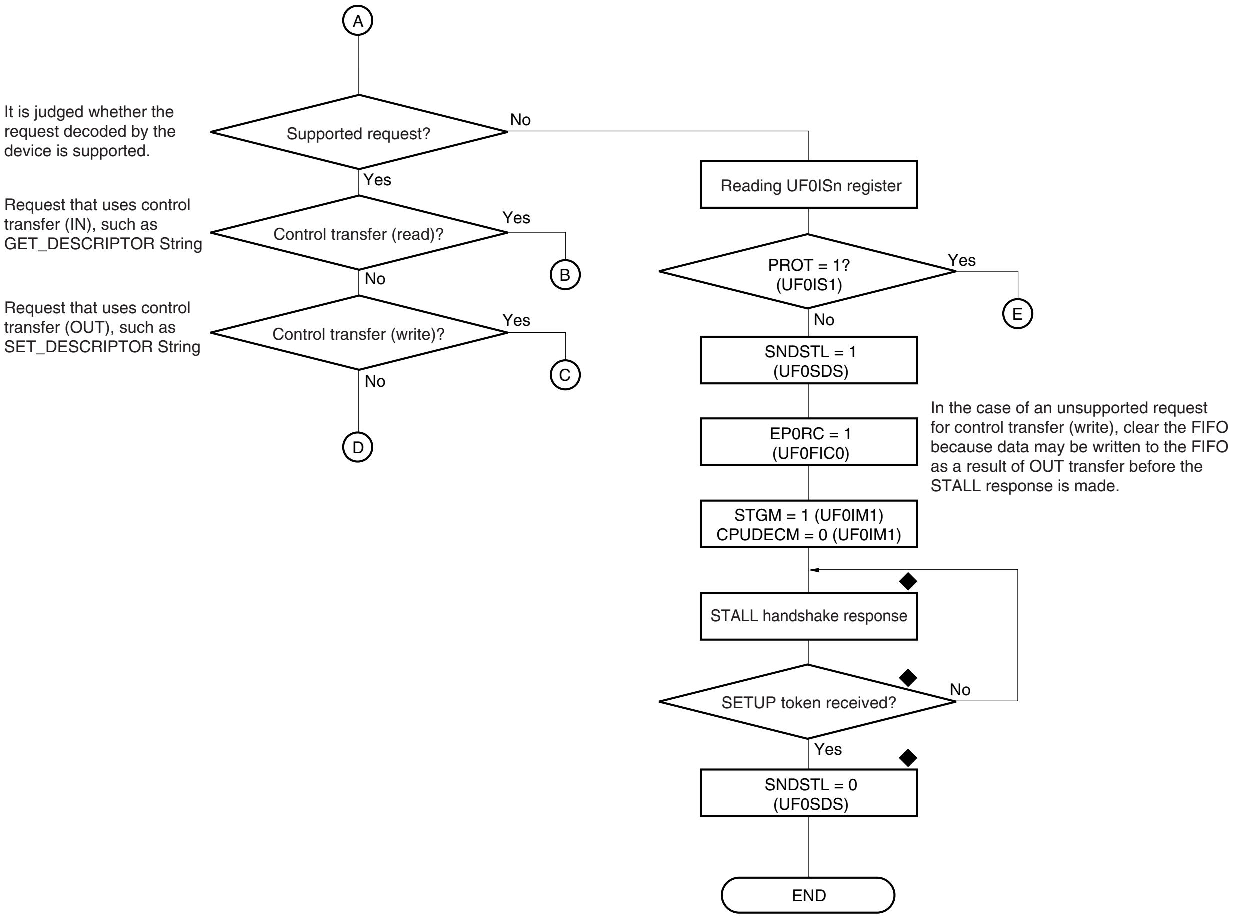


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (2/12)

(a) Token phase (2/2)

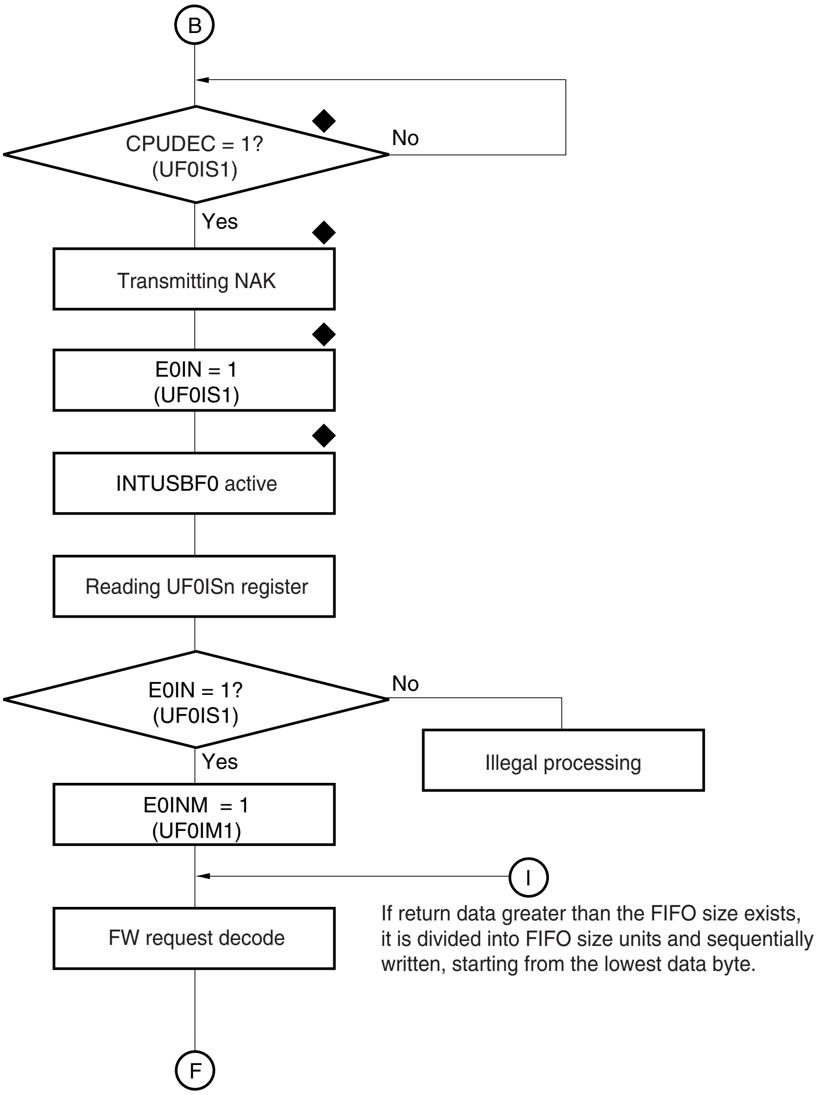

A
It is judged whether the request decoded by the device is supported.
Supported request?
No
Yes
Request that uses control transfer (IN), such as GET_DESCRIPTOR String
Control transfer (read)?
Yes
B
No
Request that uses control transfer (OUT), such as SET_DESCRIPTOR String
Control transfer (write)?
Yes
C
No
D
Reading UF0ISn register
PROT = 1? (UF0IS1)
Yes
E
No
SNDSTL = 1 (UF0SDS)
EP0RC = 1 (UF0FIC0)
In the case of an unsupported request for control transfer (write), clear the FIFO because data may be written to the FIFO as a result of OUT transfer before the STALL response is made.
STGM = 1 (UF0IM1) CPUDECM = 0 (UF0IM1)
STALL handshake response
SETUP token received?
No
Yes
SNDSTL = 0 (UF0SDS)
END

Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (3/12)

(b) Control transfer (read) (1/4)



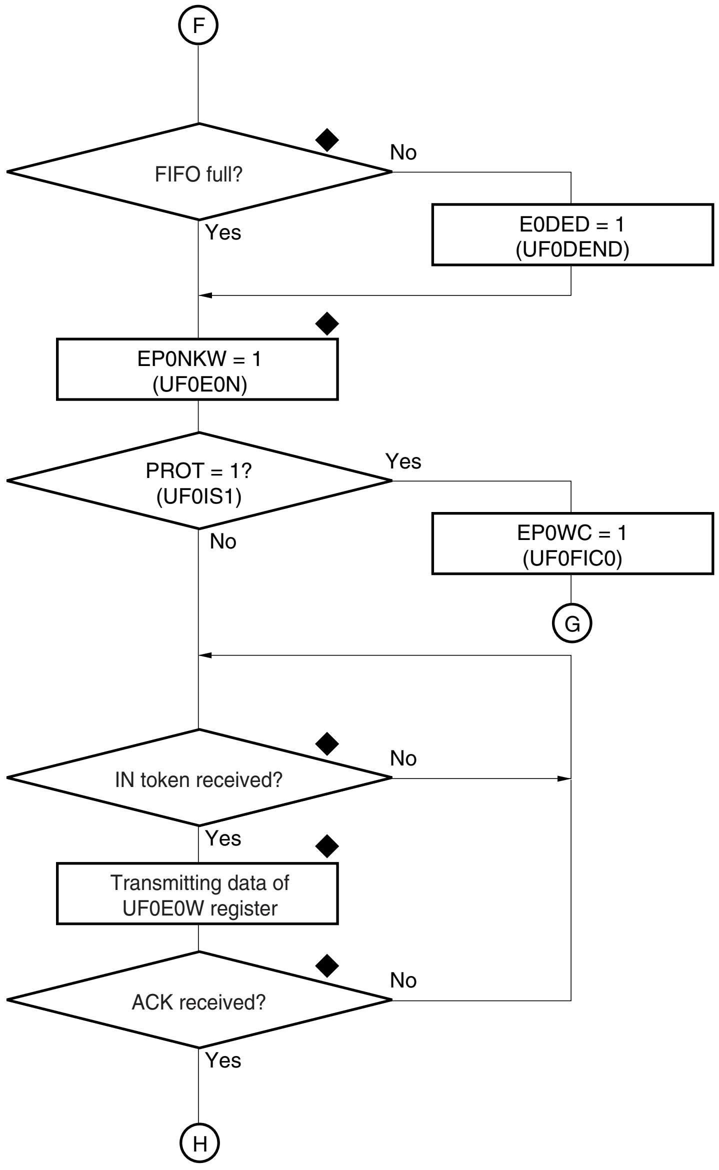
If return data greater than the FIFO size exists, it is divided into FIFO size units and sequentially written, starting from the lowest data byte.

Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24 CPUDEC Request for Control Transfer (4/12)

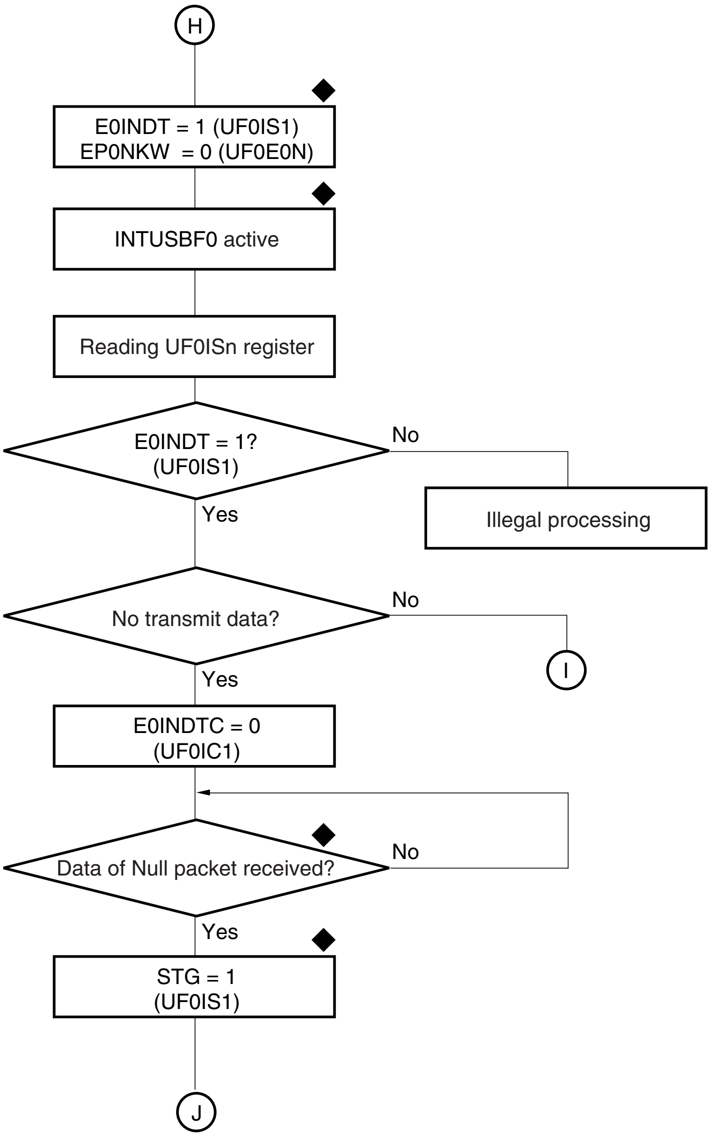
(b) Control transfer (read) (2/4)



Remark ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (5/12)

(b) Control transfer (read) (3/4)

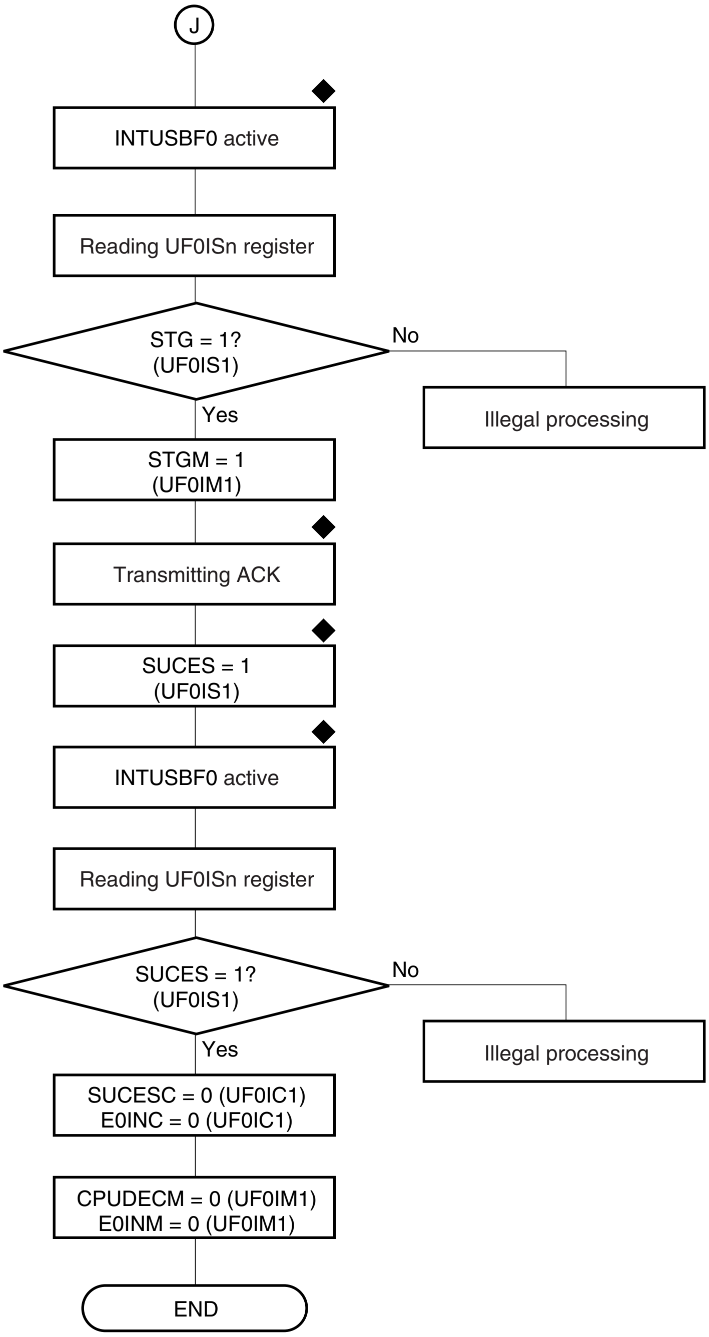


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (6/12)

(b) Control transfer (read) (4/4)

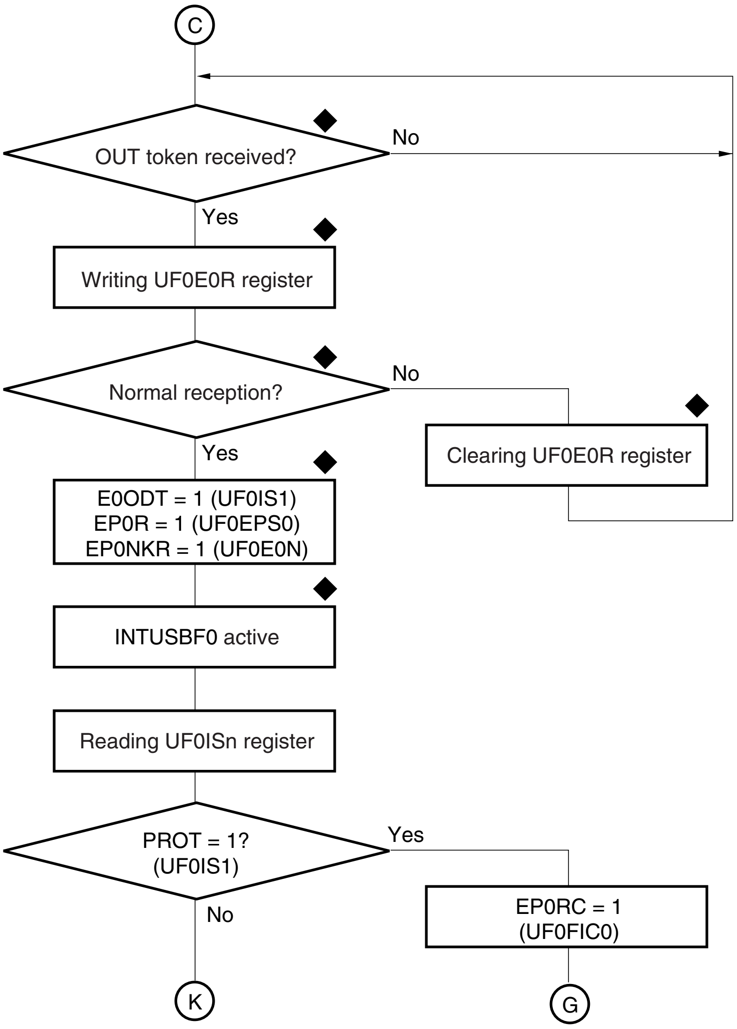


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (7/12)

(c) Control transfer (write) (1/4)

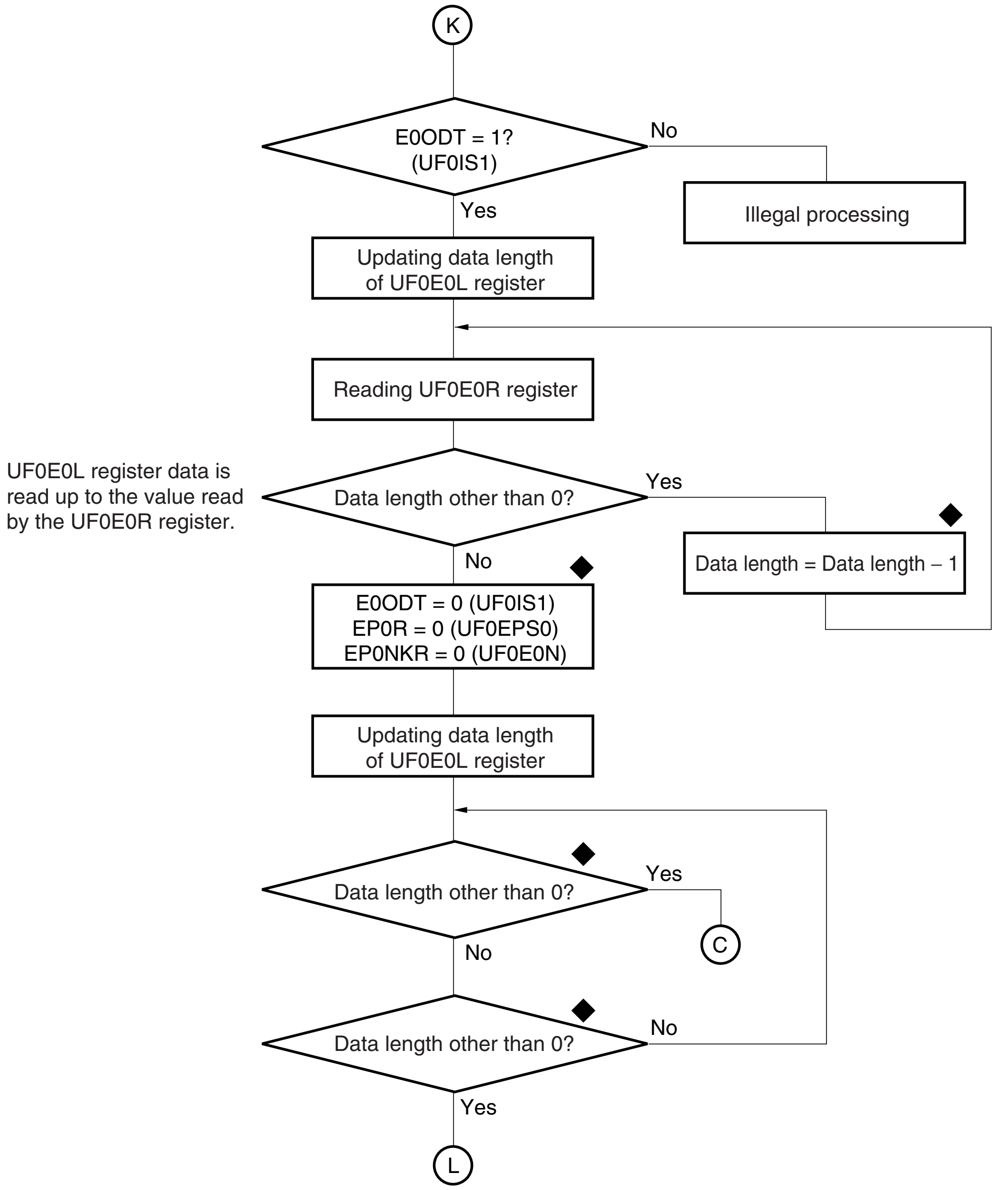


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (8/12)

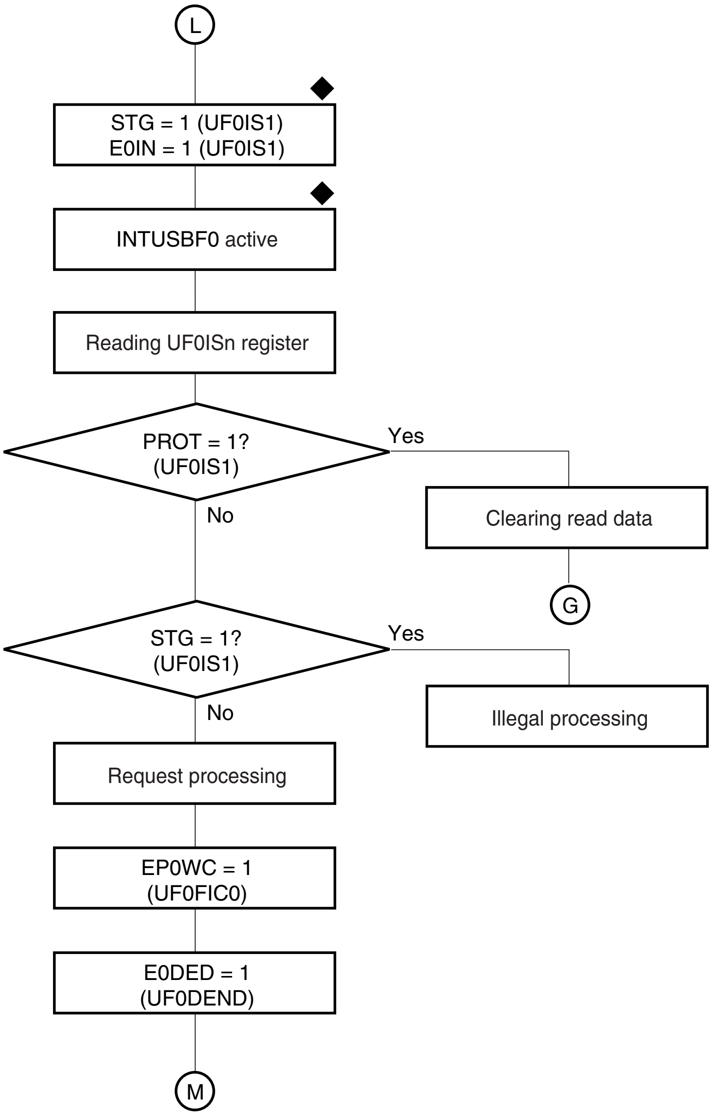
(c) Control transfer (write) (2/4)



Remark ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (9/12)

(c) Control transfer (write) (3/4)

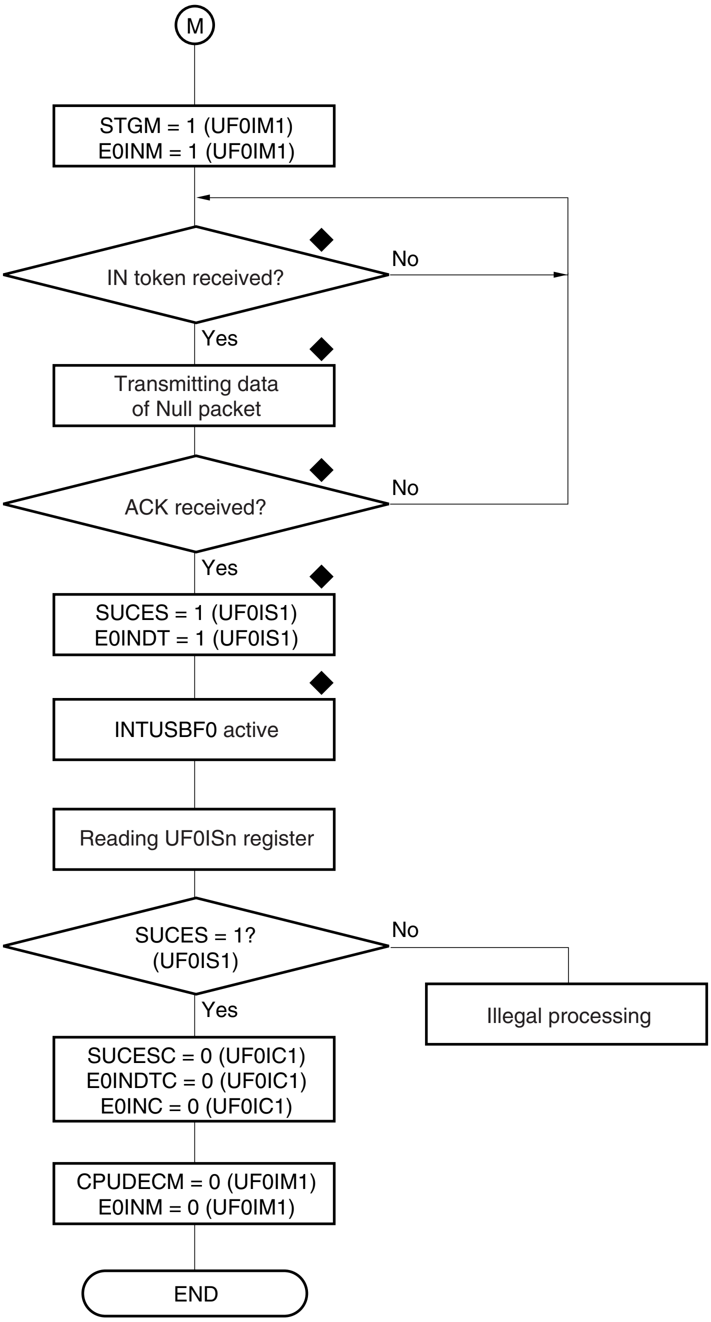


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (10/12)

(c) Control transfer (write) (4/4)

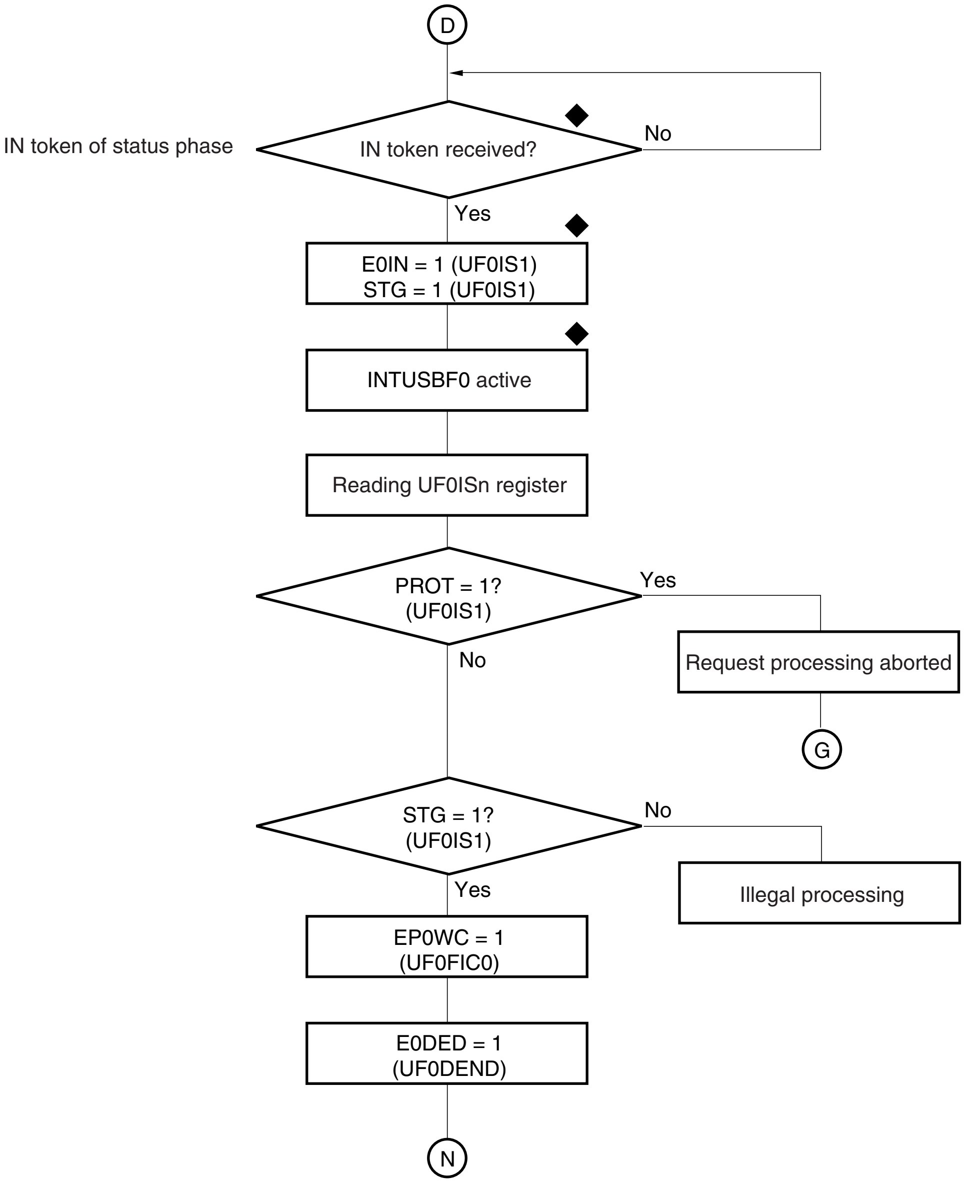


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (11/12)

(d) Control transfer (without data stage) (1/2)

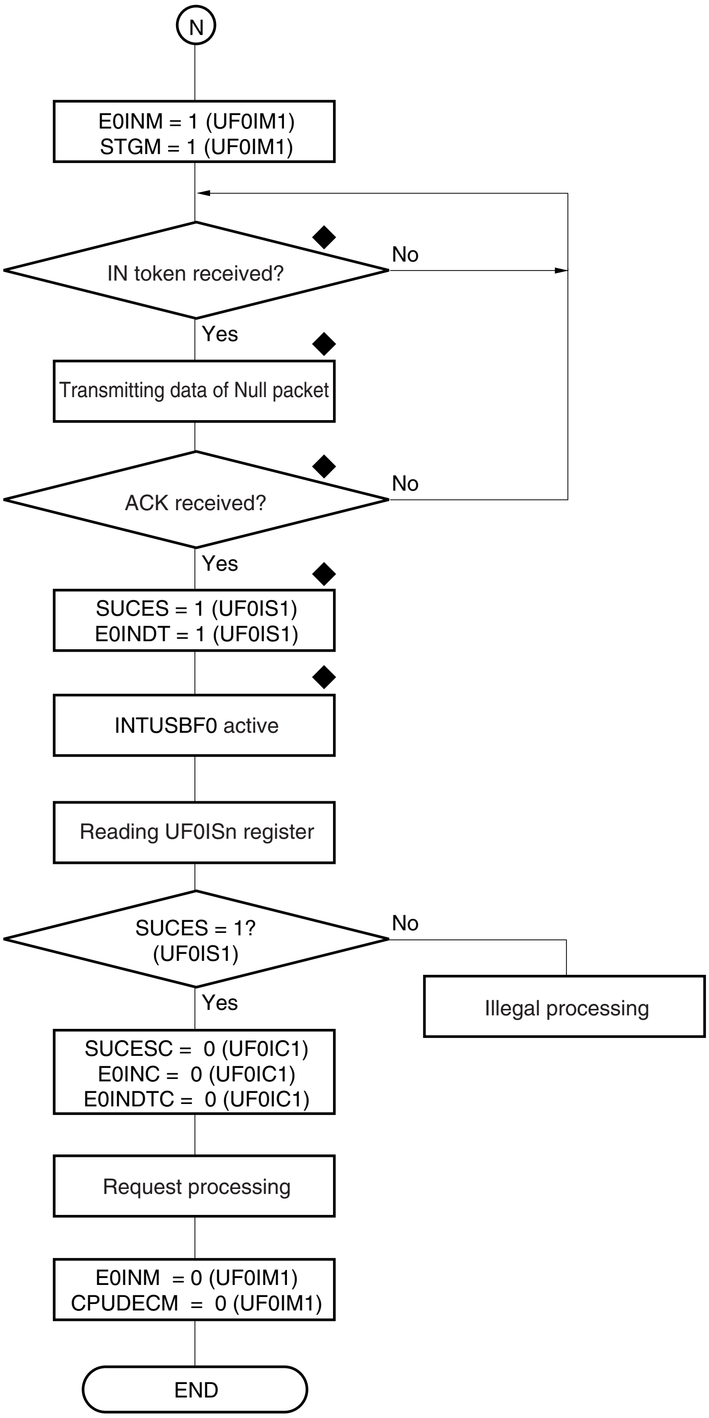


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-24. CPUDEC Request for Control Transfer (12/12)

(d) Control transfer (without data stage) (2/2)



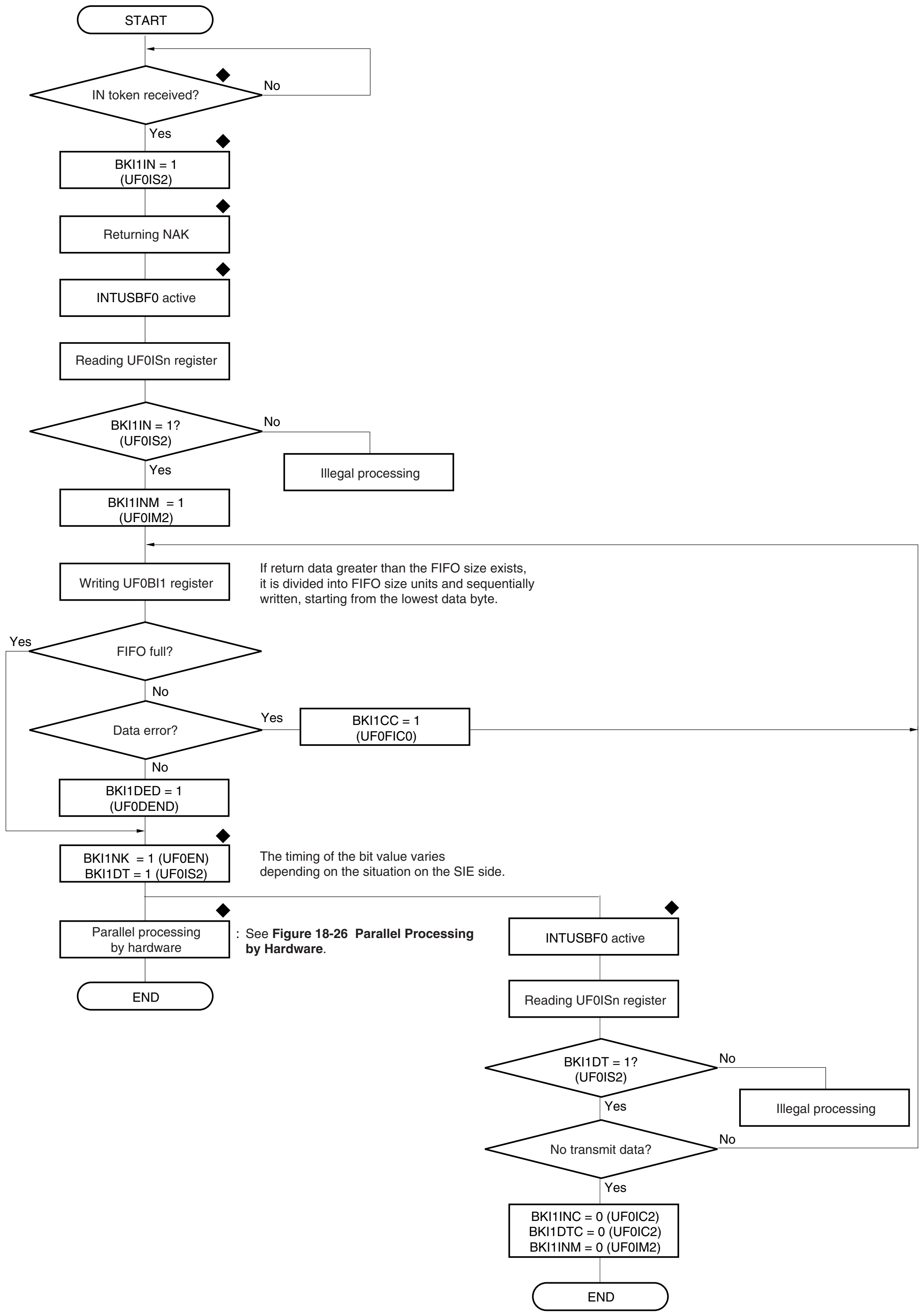
Remarks 1. n = 0, 1

2. ♦: Processing by hardware

(4) Processing for bulk transfer (IN)

Bulk transfer (IN) is allocated to Endpoint1 and Endpoint3. The flowchart shown below illustrates how Endpoint1 is controlled. Endpoint3 can also be controlled in the same sequence. To use this flowchart as the control flow of Endpoint3, therefore, read the bit names of Endpoint1 in the flowchart as those of Endpoint3.

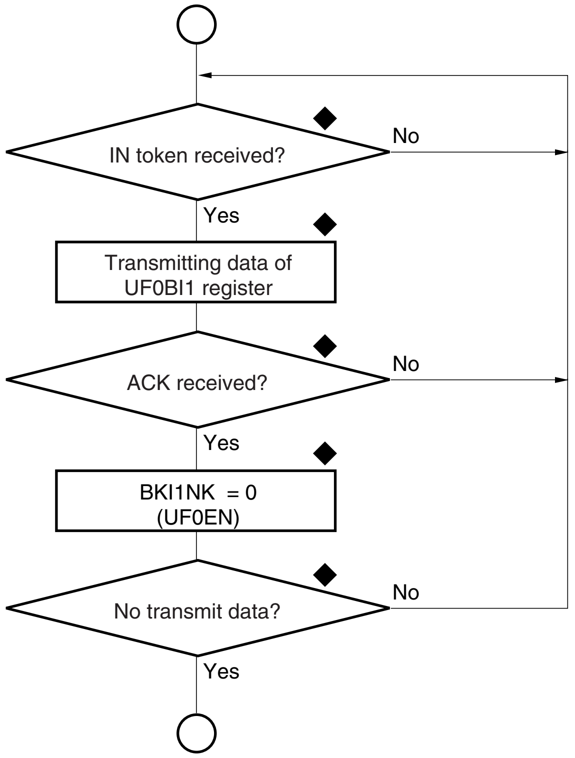
Figure 18-25. Processing for Bulk Transfer (IN) (Endpoint1)



Remarks 1. n = 2, 3

2. ♦: Processing by hardware

Figure 18-26. Parallel Processing by Hardware

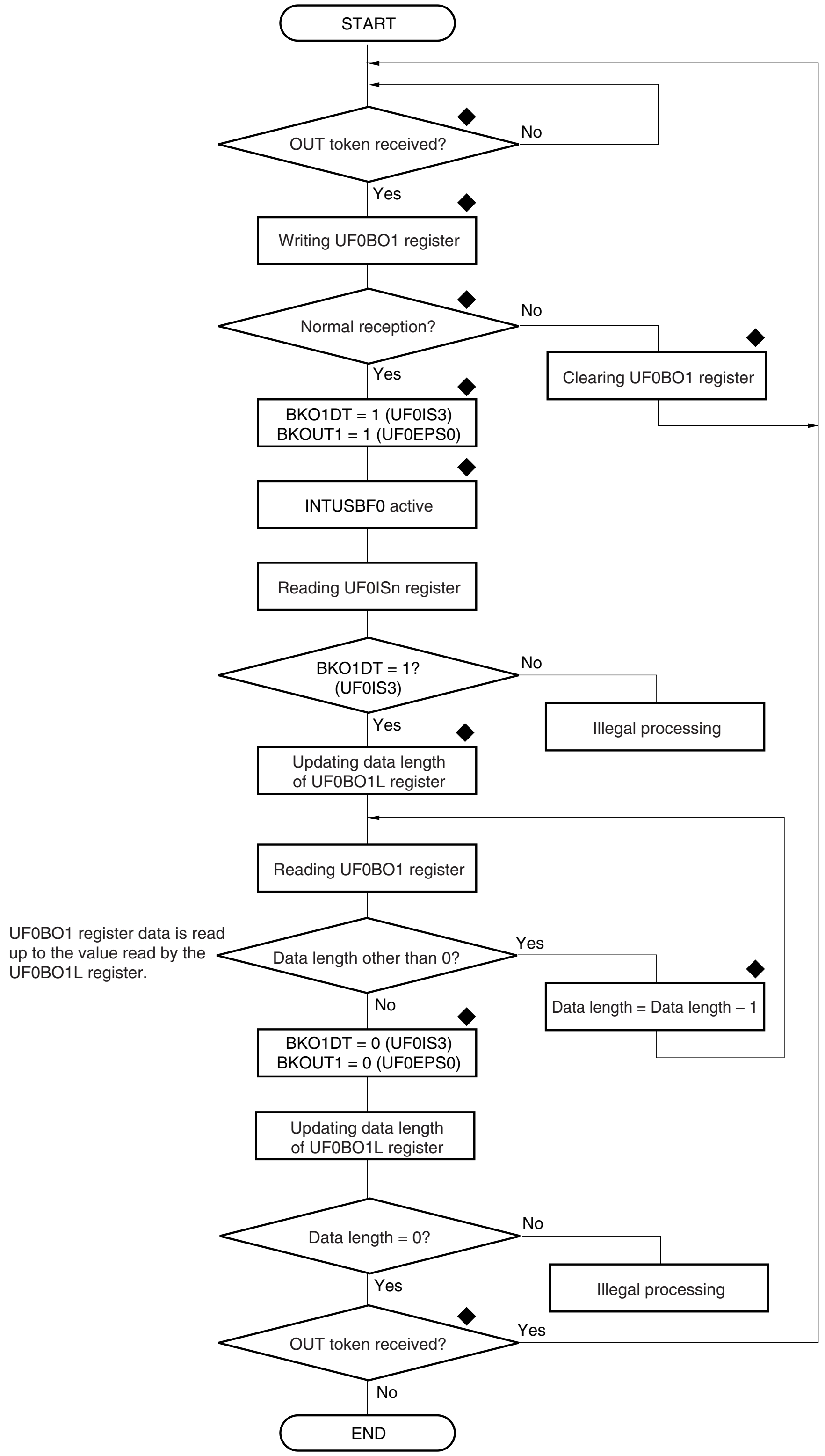


Remark ♦: Processing by hardware

(5) Processing for bulk transfer (OUT)

Bulk transfer (OUT) is allocated to Endpoint2 and Endpoint4. The flowchart shown below illustrates how Endpoint2 is controlled. Endpoint4 can also be controlled in the same sequence. To use this flowchart as the control flow of Endpoint4, therefore, read the bit names of Endpoint2 in the flowchart as those of Endpoint4.

Figure 18-27. Normal Processing for Bulk Transfer (OUT) (Endpoint2)

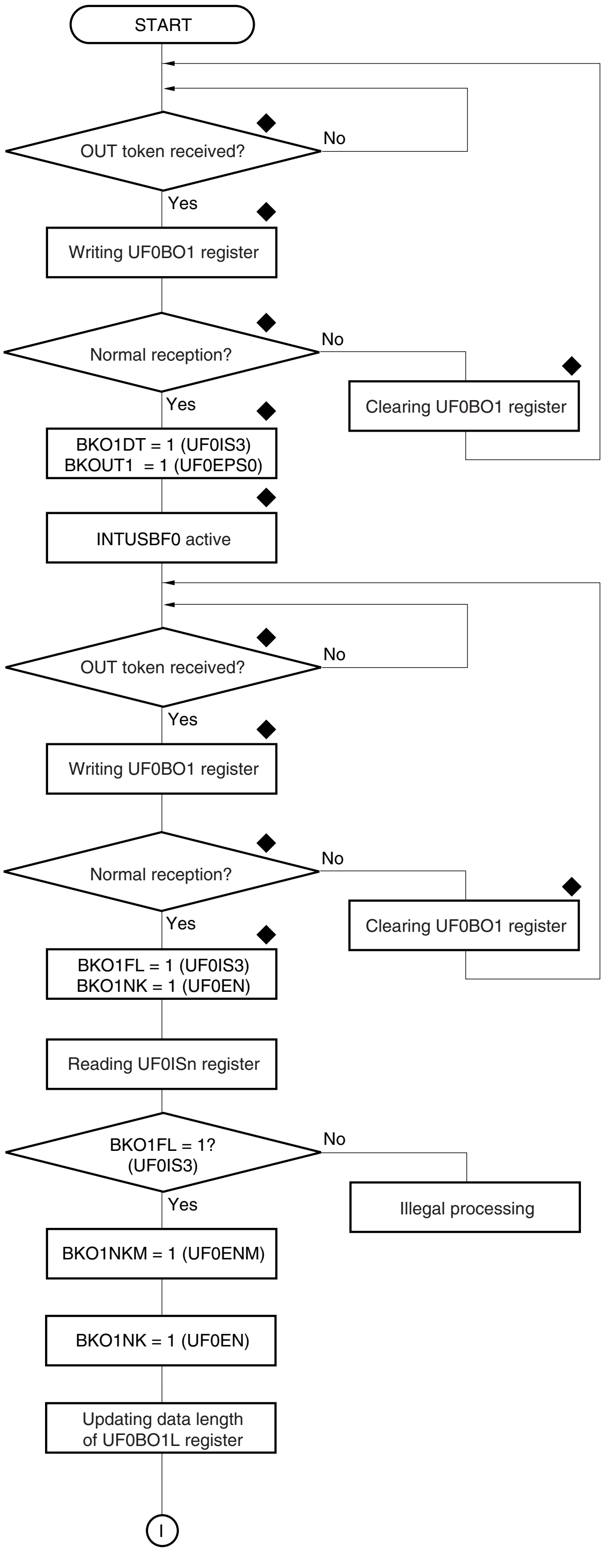


Remarks 1. n = 2, 3

2. ♦: Processing by hardware

During bulk transfer (OUT), more data may be transmitted from the host than expected by the system. Endpoint2 and Endpoint4 for bulk transfer (OUT) of the V850E/IG4-H and V850E/IH4-H consist of two 64-byte buffers so that NAK responses are suppressed as much as possible and data can be read from the CPU side even while the bus side is being accessed as the transfer rate of the USB bus increases. Consequently, if the host sends more data than expected by the system, up to 128 bytes of extra data may be automatically received in the worst case. In this case, change the control flow from that of the normal processing of Endpoint2 and Endpoint4 to the flow illustrated below when the quantity of data expected by the system has decreased to two packets. This flowchart illustrates how Endpoint2 is controlled. Endpoint4 can also be controlled in the same sequence. To use this flowchart as the control flow of Endpoint4, therefore, read the bit names of Endpoint2 in the flowchart as those of Endpoint4.

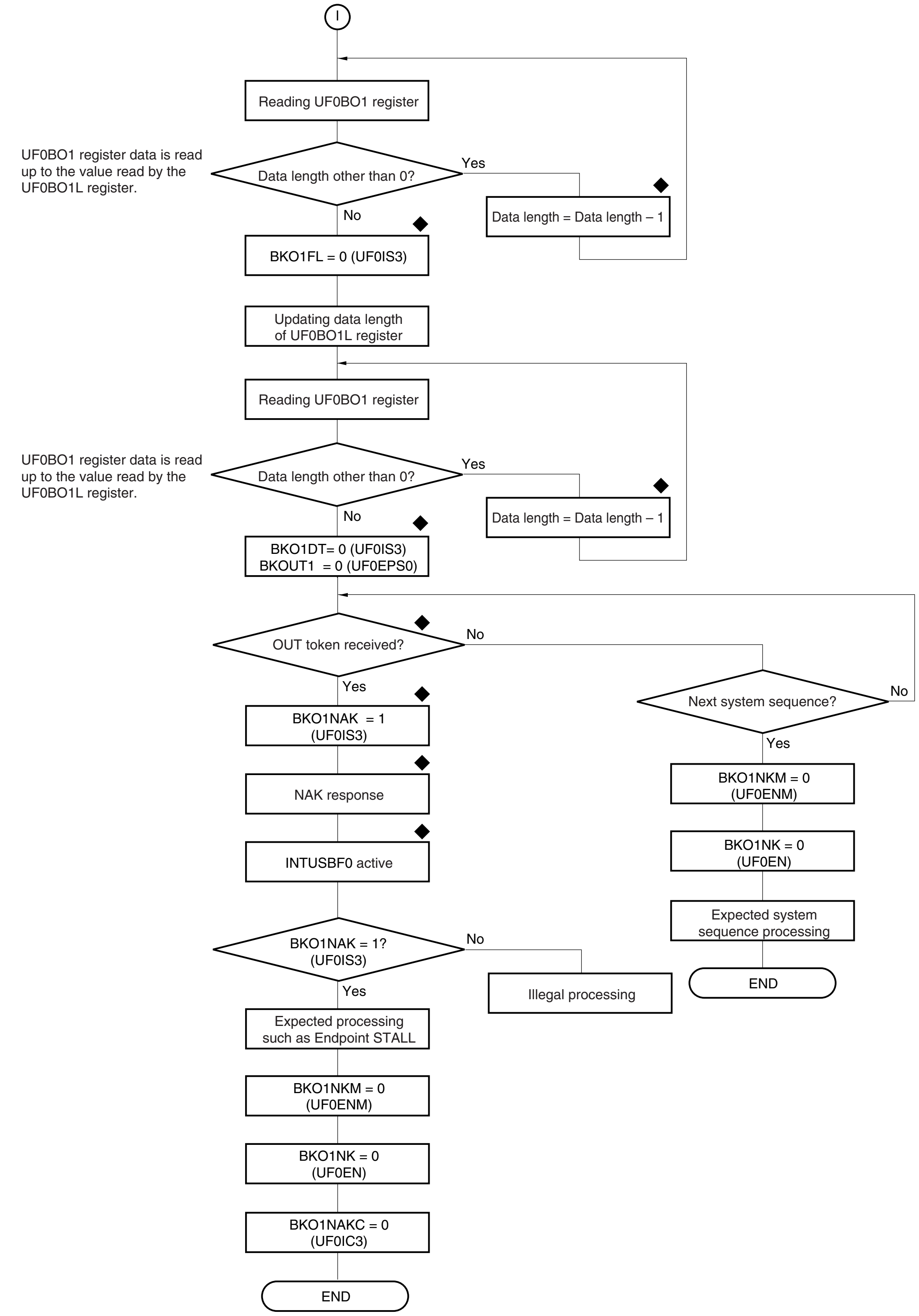
Figure 18-28. Processing If More Data Than Expected by System Is Transmitted (Endpoint2) (1/2)



Remarks 1. n = 2, 3

2. ♦: Processing by hardware

Figure 18-28. Processing If More Data Than Expected by System Is Transmitted (Endpoint2) (2/2)

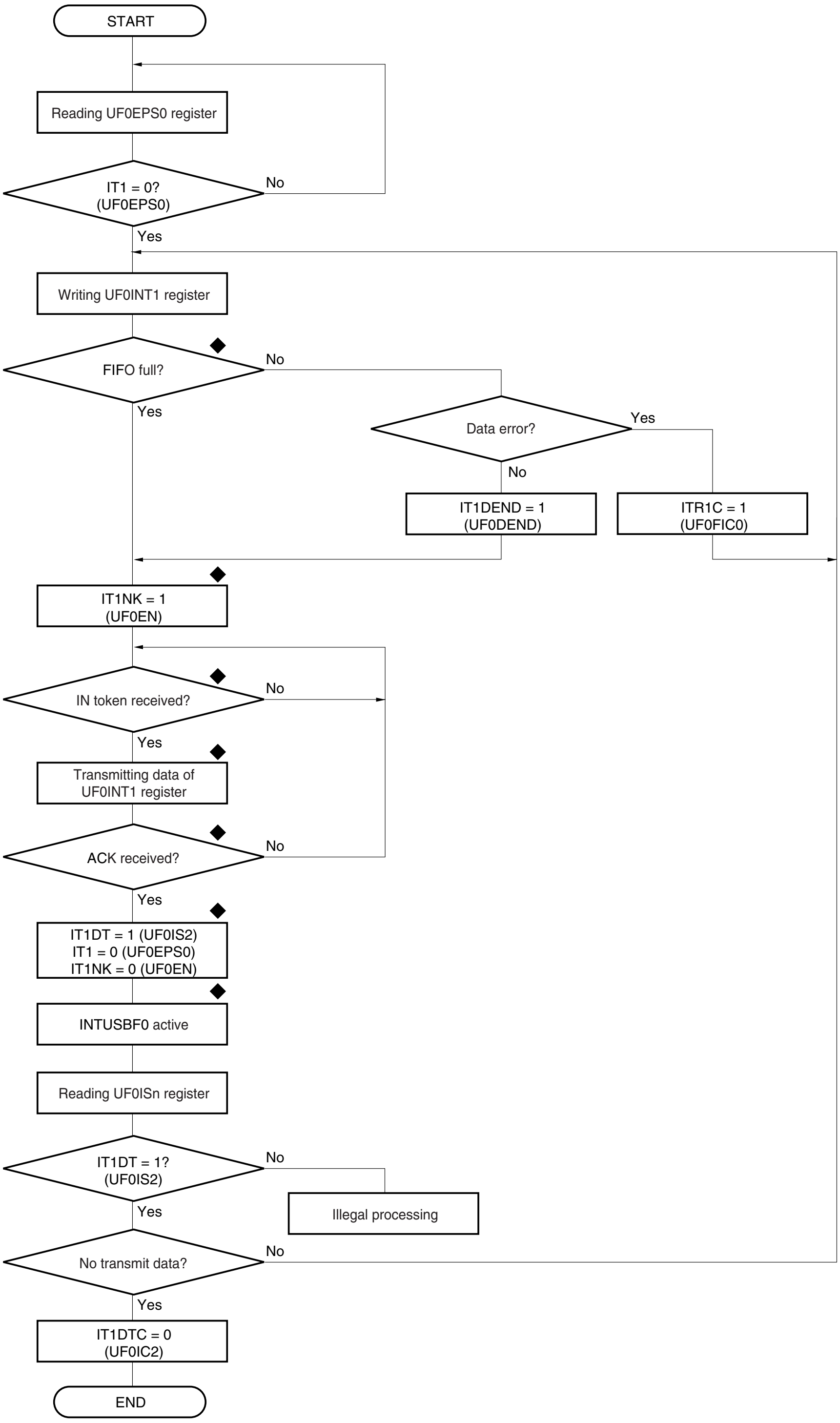

I
Reading UF0BO1 register
UF0BO1 register data is read up to the value read by the UF0BO1L register.
Data length other than 0?
Yes
Data length = Data length – 1
No
BKO1FL = 0 (UF0IS3)
Updating data length of UF0BO1L register
Reading UF0BO1 register
UF0BO1 register data is read up to the value read by the UF0BO1L register.
Data length other than 0?
Yes
Data length = Data length – 1
No
BKO1DT= 0 (UF0IS3)
BKOUT1 = 0 (UF0EPS0)
OUT token received?
No
Next system sequence?
No
Yes
BKO1NKM = 0 (UF0ENM)
BKO1NK = 0 (UF0EN)
Expected system sequence processing
END
Yes
BKO1NAK = 1 (UF0IS3)
NAK response
INTUSBF0 active
BKO1NAK = 1? (UF0IS3)
No
Illegal processing
Yes
Expected processing such as Endpoint STALL
BKO1NKM = 0 (UF0ENM)
BKO1NK = 0 (UF0EN)
BKO1NAKC = 0 (UF0IC3)
END

Remark ♦: Processing by hardware

(6) Processing for interrupt transfer (IN)

Interrupt transfer (IN) is allocated to Endpoint7. The flowchart is shown in **Figure 18-29**.

Figure 18-29. Processing for Interrupt Transfer (IN) (Endpoint7)



Remarks 1. n = 2, 3

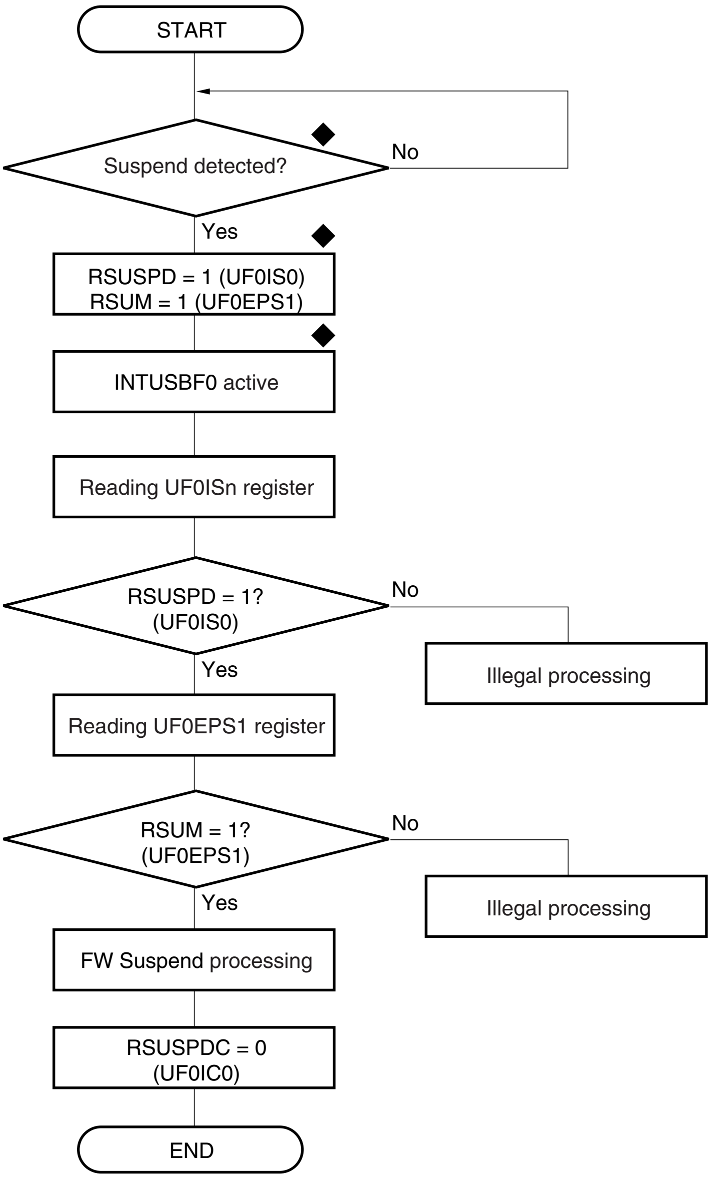
2. ♦: Processing by hardware

18.9.4 Suspend/Resume processing

How Suspend/Resume processing is performed differs depending on the configuration of the system. One example is given below.

Figure 18-30. Example of Suspend/Resume Processing (1/3)

(a) Example of Suspend processing

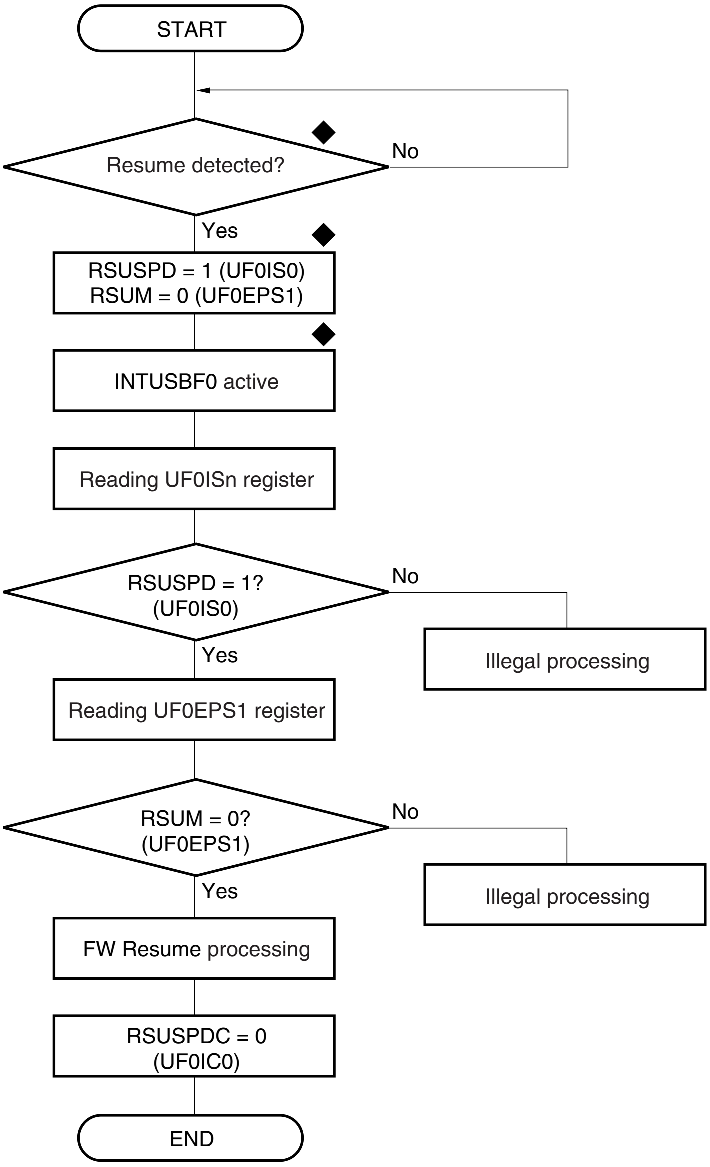


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-30. Example of Suspend/Resume Processing (2/3)

(b) Example of Resume processing

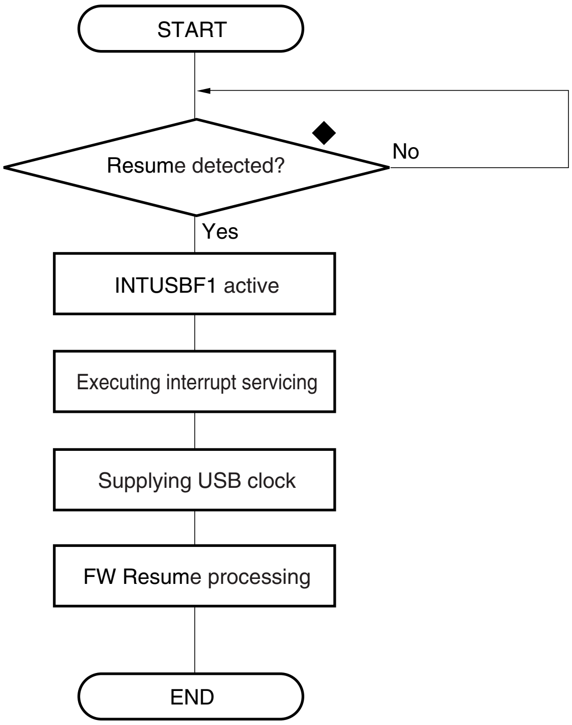


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 18-30. Example of Suspend/Resume Processing (3/3)

(c) Example of Resume processing (when supply of USB clock to USBF is stopped)



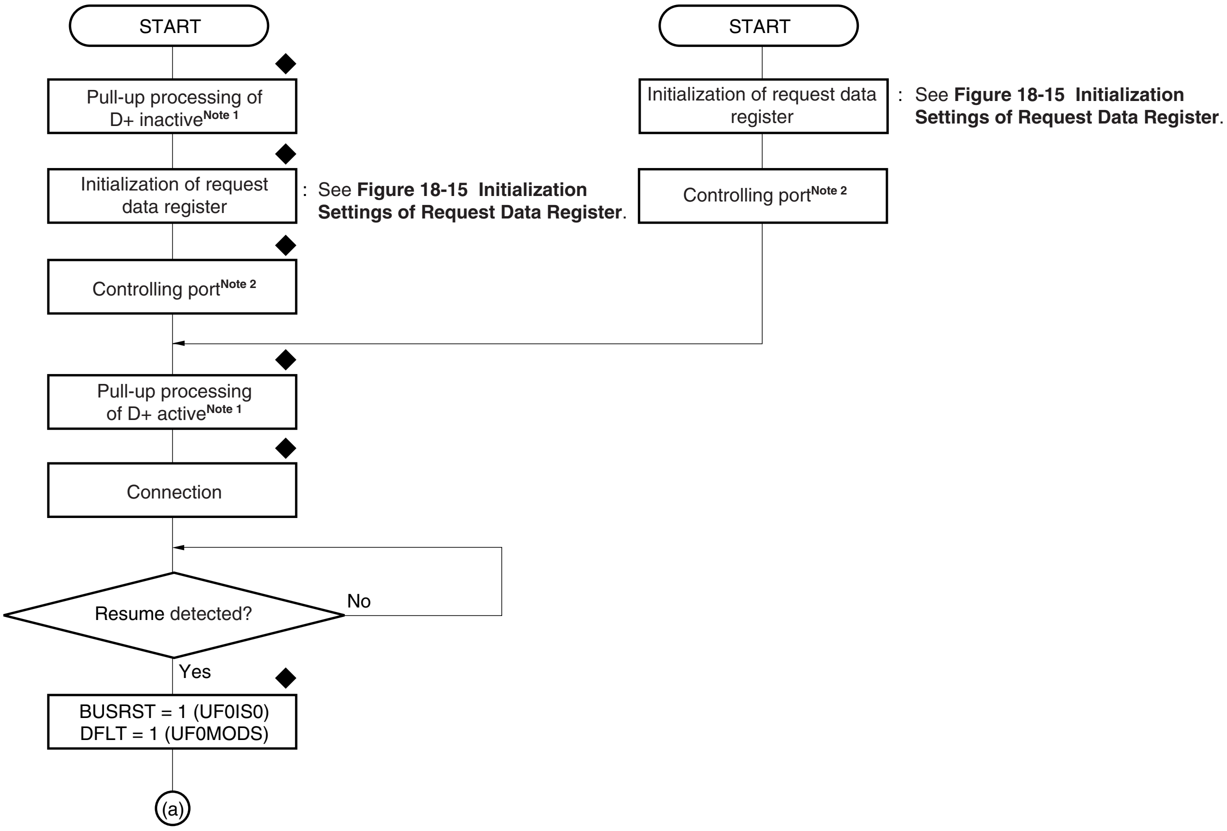
Remark ◆: Processing by hardware

18.9.5 Processing after power application

The processing to be performed after power application differs depending on the configuration of the system. One example is given below.

Figure 18-31. Example of Processing After Power Application/Power Failure (1/3)

(a) Processing after power application (1/2)


START
Pull-up processing of D+ inactive[Note 1]
Initialization of request data register
: See **Figure 18-15 Initialization Settings of Request Data Register**.
Controlling port[Note 2]
Pull-up processing of D+ active[Note 1]
Connection
Resume detected?
No
Yes
BUSRST = 1 (UF0IS0)
DFLT = 1 (UF0MODS)
(a)
START
Initialization of request data register
: See **Figure 18-15 Initialization Settings of Request Data Register**.
Controlling port[Note 2]

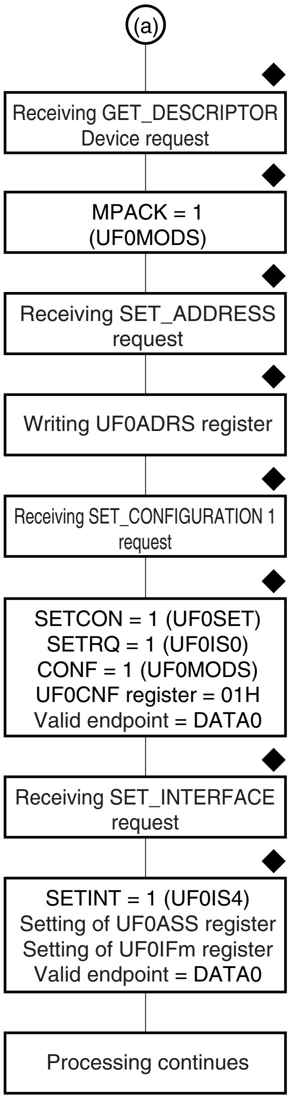
Notes 1. Use one general-purpose port pin for the signal that controls switching of the pull-up resistor of the USB bus.

2. The input mode or control mode of the general-purpose port pin allocated in **Note 1** may be selected as the default value. Note the active level of pull-up processing of D+ on power application.

Remark ♦: Processing by hardware

Figure 18-31. Example of Processing After Power Application/Power Failure (2/3)

(a) Processing after power application (2/2)

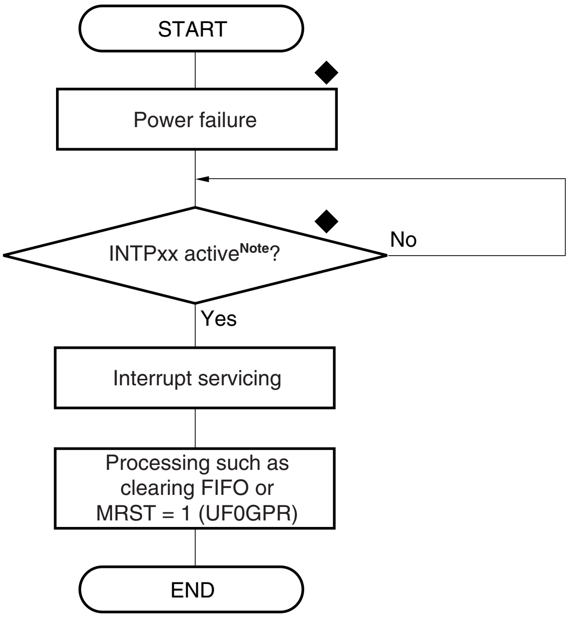


Remarks 1. m = 0 to 4

2. ♦: Processing by hardware

Figure 18-31. Example of Processing After Power Application/Power Failure (3/3)

(b) Processing on power failure



Note INTPxx indicates the external interrupt pins of the V850E/IG4-H and V850E/IH4-H (INTP00 to INTP19, INTADT0, and INTADT1), and also indicates interrupts input by the external trigger pins (TIA20, TIA21, TRGB0, TRGB1, TIT20, TIT21, TIT30, TIT31) of the timer.

Allocate one external interrupt pin to the following applications.

- Detecting disconnection of the connector in the case of self-powered mode (SFPW bit of UF0DSTL register = 1). In this case, monitor the VDD line of the USB connector, and input the result to the external interrupt pin at the edge. Note that the noise elimination time is that of the interrupt input pin, and that of each timer.
- Detecting turning off power from the HUB when the device is mounted on the same board as a HUB chip.

Remark ♦: Processing by hardware

CHAPTER 19 BUS CONTROL FUNCTION

The V850E/IG4-H and V850E/IH4-H are provided with an external bus interface function via which external memories such as ROM and RAM, and I/O devices can be connected to areas other than the internal ROM, internal RAM, or on-chip peripheral I/O registers via ports 0, 2 to 4, 9 (V850E/IH4-H only), and DL. These ports control address/data I/O, the read/write strobe signal, waits, the clock output, and the address strobe signal.

19.1 Features

- 16-bit/8-bit data bus sizing function
- 3-space chip select function
  (The $\overline{\text{CS2}}$ signal does not exit as the external signal of the V850E/IG4-H and V850E/IH4-H. This signal is used as the chip select signal for the USB function area in the V850E/IG4-H and V850E/IH4-H.)
- Wait function
  - Programmable wait function, through which up to 7 wait states can be inserted for each memory block
  - Address setup wait and address hold wait insertion functions, through which 1 wait state can be inserted for each memory block
  - External wait function via $\overline{\text{WAIT}}$ pin
- Idle state insertion function
  - A low-speed device can be connected by inserting an idle state after a read cycle.
- Bus mode
  - V850E/IG4-H: Multiplexed bus mode
  - V850E/IH4-H: Multiplexed bus mode/separate bus mode
- Support for little endian
- External bus clock frequency: (f_{BUS}) = f_{CLK}/4
- Misaligned access is possible.
- Up to 8 MB of physical memory can be connected (512 KB are shared with the internal ROM area).

19.2 Bus Control Pins

The pins used to connect an external device are listed in the table below.

Table 19-1. Bus Control Pins (When Separate Bus Mode Selected (V850E/IH4-H only))

Bus Control Pin	I/O	Function	Alternate-Function Pin	Register to Switch Between Port Mode/ Alternate-Function Mode
AD0 to AD15	I/O	Data bus	PDL0 to PDL15	PMCDL register
A0 to A7	Output	Address bus	P90 to P97	PMC9 register
$\overline{\text{WAIT}}$	Input	External wait control	P31	PMC3 register
CLKOUT	Output	External bus clock output	P07	PMC0 register
$\overline{\text{CS0}}$ to $\overline{\text{CS1}}$	Output	Chip select	P34, P32	PMC3 register
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	Output	Write strobe signal	P27, P30	PMC2 and PMC3 registers
$\overline{\text{RD}}$	Output	Read strobe signal	P44	PMC4 register
ASTB	Output	Address strobe signal	P37	PMC3 register

Table 19-2. Bus Control Pins (When Multiplexed Bus Mode Selected)

Bus Control Pin	I/O	Function	Alternate-Function Pin	Register to Switch Between Port Mode/ Alternate-Function Mode
AD0 to AD15	I/O	Address/data bus	PDL0 to PDL15	PMCDL register
$\overline{\text{WAIT}}$	Input	External wait control	P31	PMC3 register
CLKOUT	Output	External bus clock output	P07	PMC0 register
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	Output	Chip select	P34, P32	PMC3 register
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	Output	Write strobe signal	P27, P30	PMC2 and PMC3 registers
$\overline{\text{RD}}$	Output	Read strobe signal	P44	PMC4 register
ASTB	Output	Address strobe signal	P37	PMC3 register

19.2.1 Pin status during internal ROM, internal RAM, and on-chip peripheral I/O access

The status of each pin is as follows when the internal ROM, internal RAM, and on-chip peripheral I/O are accessed.

Table 19-3. Pin Status List in Internal ROM, Internal RAM, and On-Chip Peripheral I/O Access

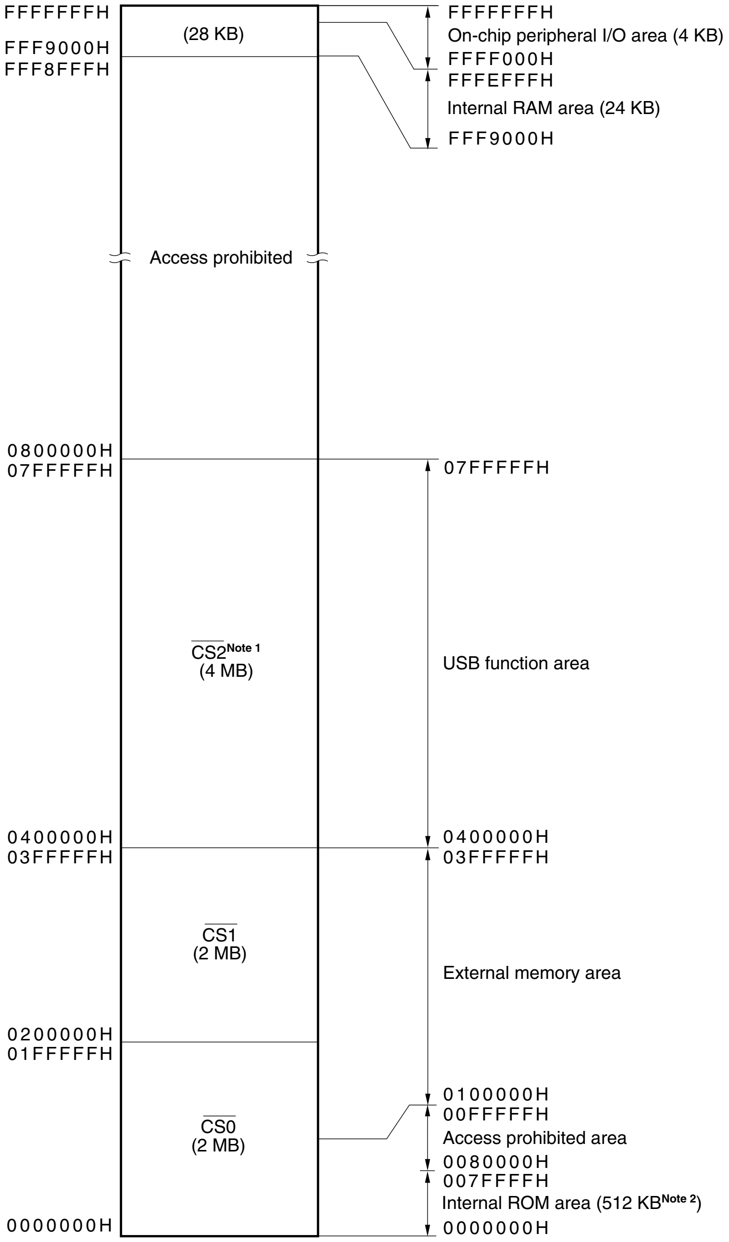
Access Destination	Internal ROM	Internal RAM	On-Chip Peripheral I/O
Address bus	Undefined	Undefined	**Note 1**
Data bus	Hi-Z	Hi-Z	Hi-Z
External bus control signal	Inactive$^{\text{Note 2}}$	Inactive$^{\text{Note 2}}$	Inactive$^{\text{Note 2}}$

Notes 1. While the on-chip peripheral I/O is accessed, the address the on-chip peripheral I/O accesses is also output to the external address bus.

2. The $\overline{\text{WAIT}}$ pin does not input any signal during this operation.

19.3 Memory Block Function

The lower 8 MB of the 256 MB memory space is reserved for external memory expansion and is divided into memory blocks of 2 MB, 2 MB, and 4 MB. The bus width and programmable wait function can be independently specified for each block.



Notes 1. $\overline{\text{CS2}}$ signal is not provided as an external signal of the V850E/IG4-H and V850E/IH4-H.
$\overline{\text{CS2}}$ is used as the chip select signal for the USB function area in the internal V850E/IG4-H and V850E/IH4-H.

2.

μPD70F3919, 70F3922:	256 KB
μPD70F3920, 70F3923:	384 KB
μPD70F3921, 70F3924:	480 KB

19.3.1 Chip select control function

Of the 256 MB address space (linear), the lower 8 MB (0000000H to 07FFFFFH) has three chip select functions, $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$. The areas selected by $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$ are fixed.

The memory area can be effectively used by dividing it into memory blocks using the chip select control function. The allocation of memory blocks is described below.

Table 19-4. Chip Select Area

Chip Select Signal	Area
$\overline{\text{CS0}}$	0000000H to 01FFFFFH (2 MB)
$\overline{\text{CS1}}$	0200000H to 03FFFFFH (2 MB)
$\overline{\text{CS2}}$ Note	0400000H to 07FFFFFH (4 MB)

Note $\overline{\text{CS2}}$ signal is not provided as an external signal of the V850E/IG4-H and V850E/IH4-H. $\overline{\text{CS2}}$ is used as the chip select signal for the USB function area in the internal V850E/IG4-H and V850E/IH4-H.

19.4 Bus Cycle Type Control Function

In the V850E/IG4-H and V850E/IH4-H, SRAM, external ROM, and external I/O can be connected directly.

(1) Bus cycle type configuration register 0 (BCT0)

This register can be read or written in 16-bit units.
Reset sets this register to CCCCH.

Cautions 1. Do not access an external memory area until the initial setting of the BCT0 register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

2. The set contents of each register are invalid for the chip select space where operations are prohibited.

After reset: CCCCH R/W Address: FFFFF480H

	15	14	13	12	11	10	9	8
BCT0	1	1	0	0	1	1	0	0
$\overline{\text{CSn}}$ signal								

	7	6	5	4	3	2	1	0
	ME1	1	0	0	ME0	1	0	0
$\overline{\text{CSn}}$ signal	$\overline{\text{CS1}}$				$\overline{\text{CS0}}$			

MEn	Memory controller operation enable for chip select space (n = 0, 1)
0	Operation disabled
1	Operation enabled

Caution Be sure to set bits 0, 1, 4, 5, 8, 9, 12, and 13 to "0", and set bits 2, 6, 10, 11, 14, and 15 to "1". If they are set other than above, the operation is not guaranteed.

19.5 Bus Access

19.5.1 Number of access clocks

The number of base clocks (MIN. value) necessary for accessing each resource is as follows.

Table 19-5. Number of Access Clocks

Bus Cycle Configuration Resource (Bus Width)		Instruction Fetch (Normal Access)	Instruction Fetch (Branch)	Operand Data Access
Internal ROM (32 bits)		1	4	7
Internal RAM (32 bits)		1[Note 2]	1[Note 3]	1
On-chip peripheral I/O (16 bits)		–	–	3 + m
External memory (16 bits)	Separate bus mode[Note 1]	3 + n	3 + n	3 + n
	Multiplexed bus mode	3 + n	3 + n	3 + n

Notes 1. V850E/IH4-H only

2. This value is 2 if there is conflict with data access.

Remarks 1. Unit: Clock/access

2. m: Number of wait states set by VSWC register
n: Number of wait states inserted

19.5.2 Bus sizing function

The bus sizing function controls the data bus width for chip select space. The data bus width is specified by using the BSC register.

If a 16-bit bus width is specified, the lower 8 bits are used for even addresses and the higher 8 bits are used for odd addresses.

(1) Bus size configuration register (BSC)

This register controls the bus width of the chip select space.
This register can be read or written in 16-bit units.
Reset sets this register to 5555H.

Caution Write to the BSC register after reset, and then do not change the set value. Also, when changing the initial values of the BSC register, do not access an external memory area until the settings are complete. However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 5555H R/W Address: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1
$\overline{\text{CSn}}$ signal								

	7	6	5	4	3	2	1	0
	0	1	0	1	0	BS10	0	BS00
$\overline{\text{CSn}}$ signal						$\overline{\text{CS1}}$		$\overline{\text{CS0}}$

BSn0	Specification of data bus width of chip select space (n = 0, 1)
0	8 bits
1	16 bits

Caution Be sure to set bits 1, 3, 5, 7, 9, 11, 13, and 15 to "0", and set bits 4, 6, 8, 10, 12, and 14 to "1". If they are set other than above, the operation is not guaranteed.

19.5.3 Endian function

The V850E/IG4-H and V850E/IH4-H support little endian.

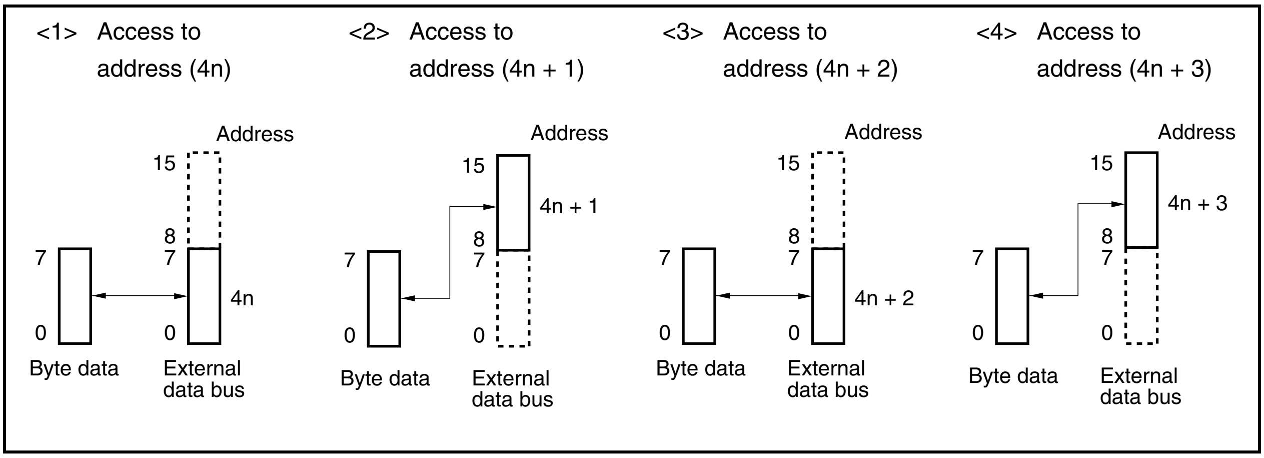
19.5.4 Bus width

The V850E/IG4-H and V850E/IH4-H access on-chip peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each type of access. All data is accessed in order starting from the lower order side.

(1) Byte access (8 bits)

(a) When the data bus width is 16 bits

8-bit data is transmitted/received via a 16-bit bus. Therefore, if an even address is specified, the lower byte of the external data bus address is accessed. If an odd address is specified, the higher byte of the external data bus address is accessed.



<1> Access to address (4n)

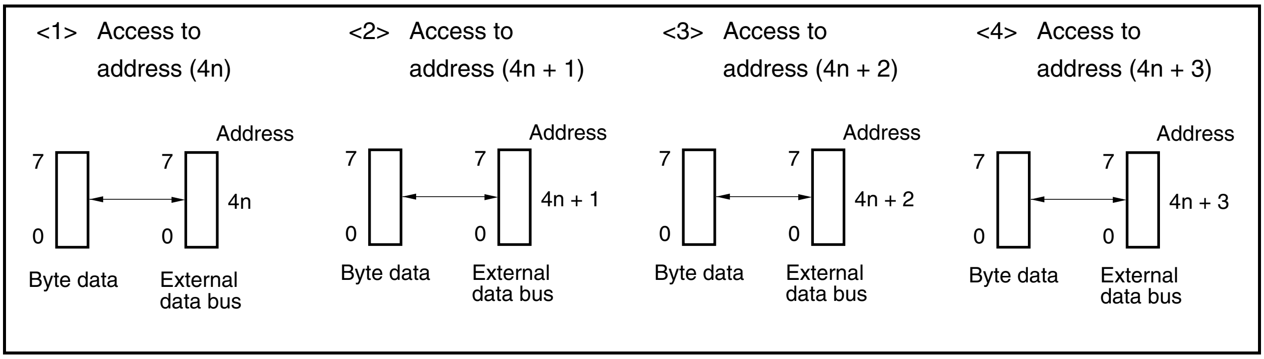
<2> Access to address (4n + 1)

<3> Access to address (4n + 2)

<4> Access to address (4n + 3)

(b) When the data bus width is 8 bits

8-bit data is transmitted/received via an 8-bit bus. Therefore, the specified even/odd address of the external data bus is accessed.



<1> Access to address (4n)

<2> Access to address (4n + 1)

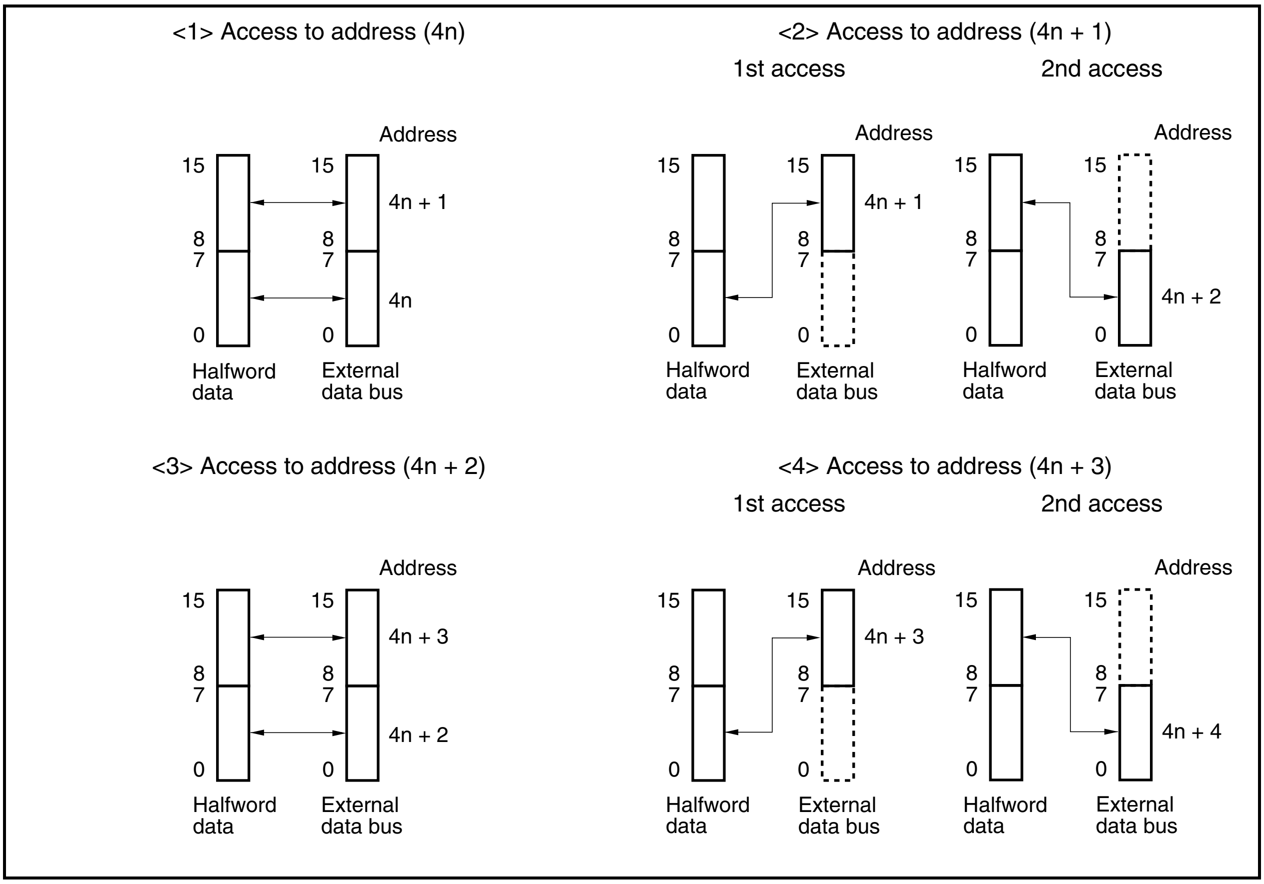
<3> Access to address (4n + 2)

<4> Access to address (4n + 3)

(2) Halfword access (16 bits)

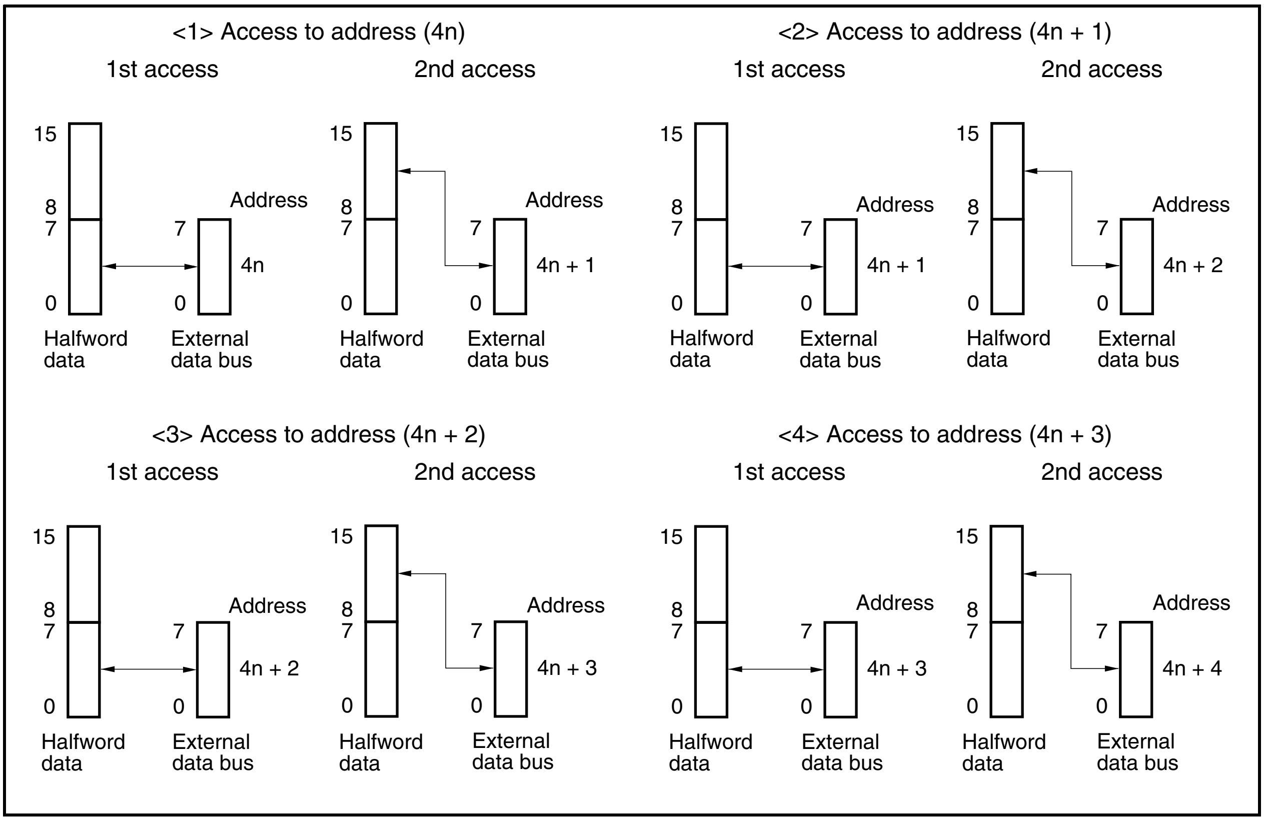
(a) When the data bus width is 16 bits

16-bit data is transmitted/received via a 16-bit bus. Therefore, if an even address is specified, the lower and higher bytes of the external data bus address are accessed at the same time. If an odd address is specified, the lower byte of the data is transmitted/received to/from an odd address via the higher byte of the external data bus address in the first access. In the second access, the higher byte of the data is transmitted/received to/from an odd address via the lower byte of the external data bus address.



(b) When the data bus width is 8 bits

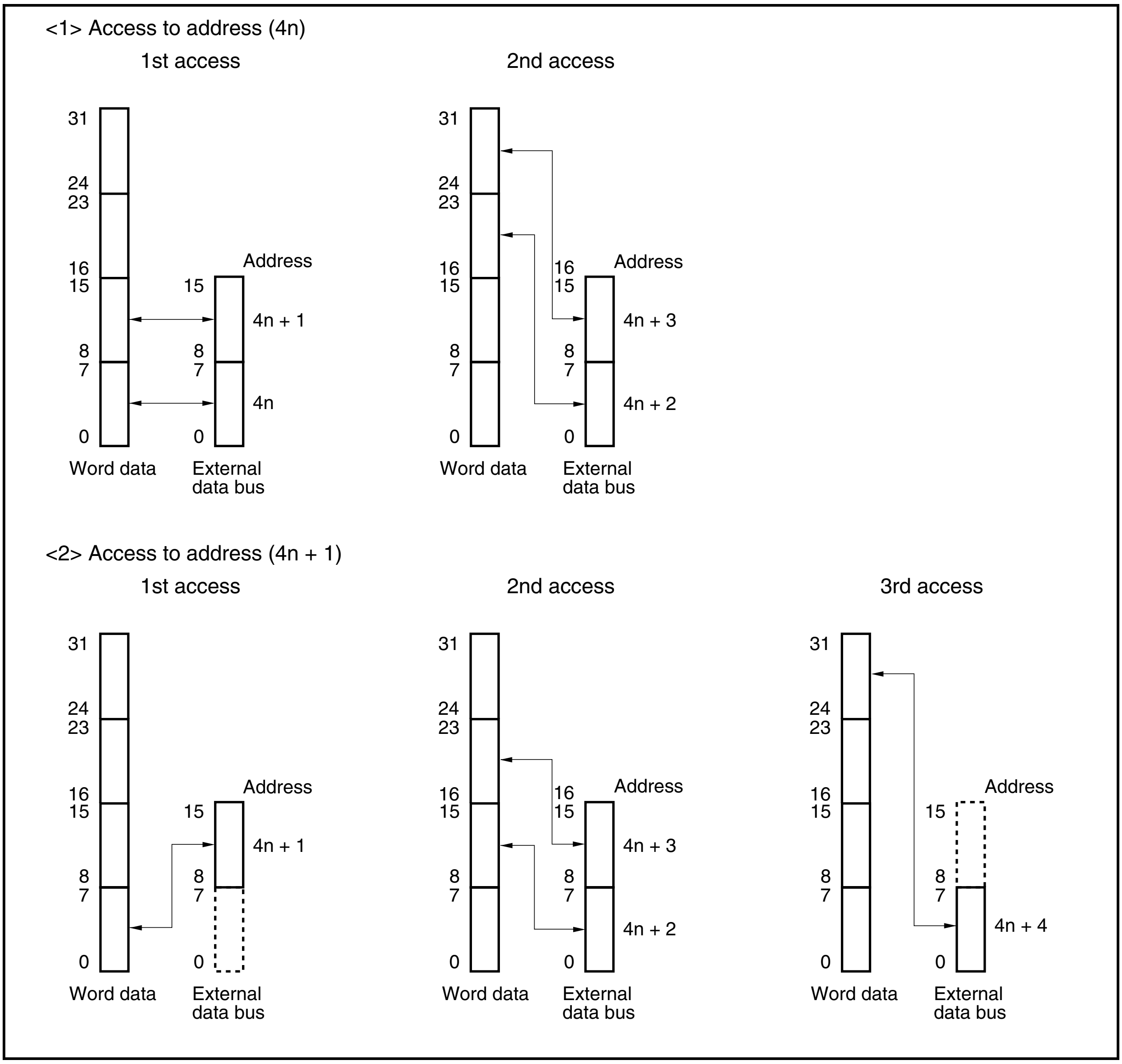
16-bit data is transmitted/received via an 8-bit bus. Therefore, the data is transmitted/received in two accesses. The lower/higher byte of the data is transmitted/received to/from the corresponding lower/higher byte of the external bus address.



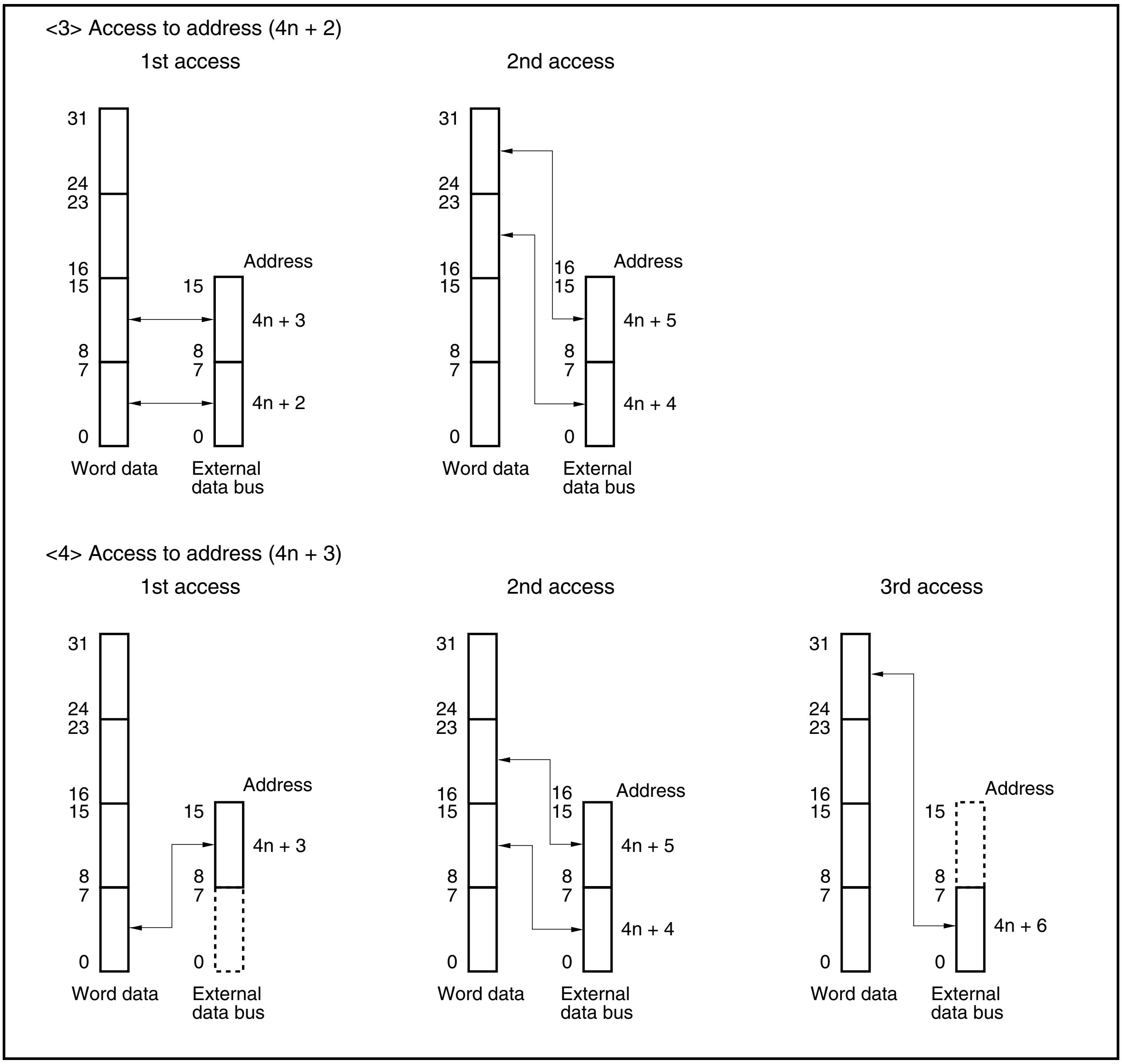
(3) Word access (32 bits)

(a) When the data bus width is 16 bits (1/2)

32-bit data is transmitted/received via a 16-bit bus. Therefore, if an even address is specified, the data is transmitted/received in two accesses in 16-bit units. If an odd address is specified, the lower quarter-word data is transmitted/received to/from the higher byte (first access), the middle halfword data is transmitted/received to/from the middle bytes (second access), and the upper quarter-word data is transmitted/received to/from the lower byte (third access), of the external data bus address.

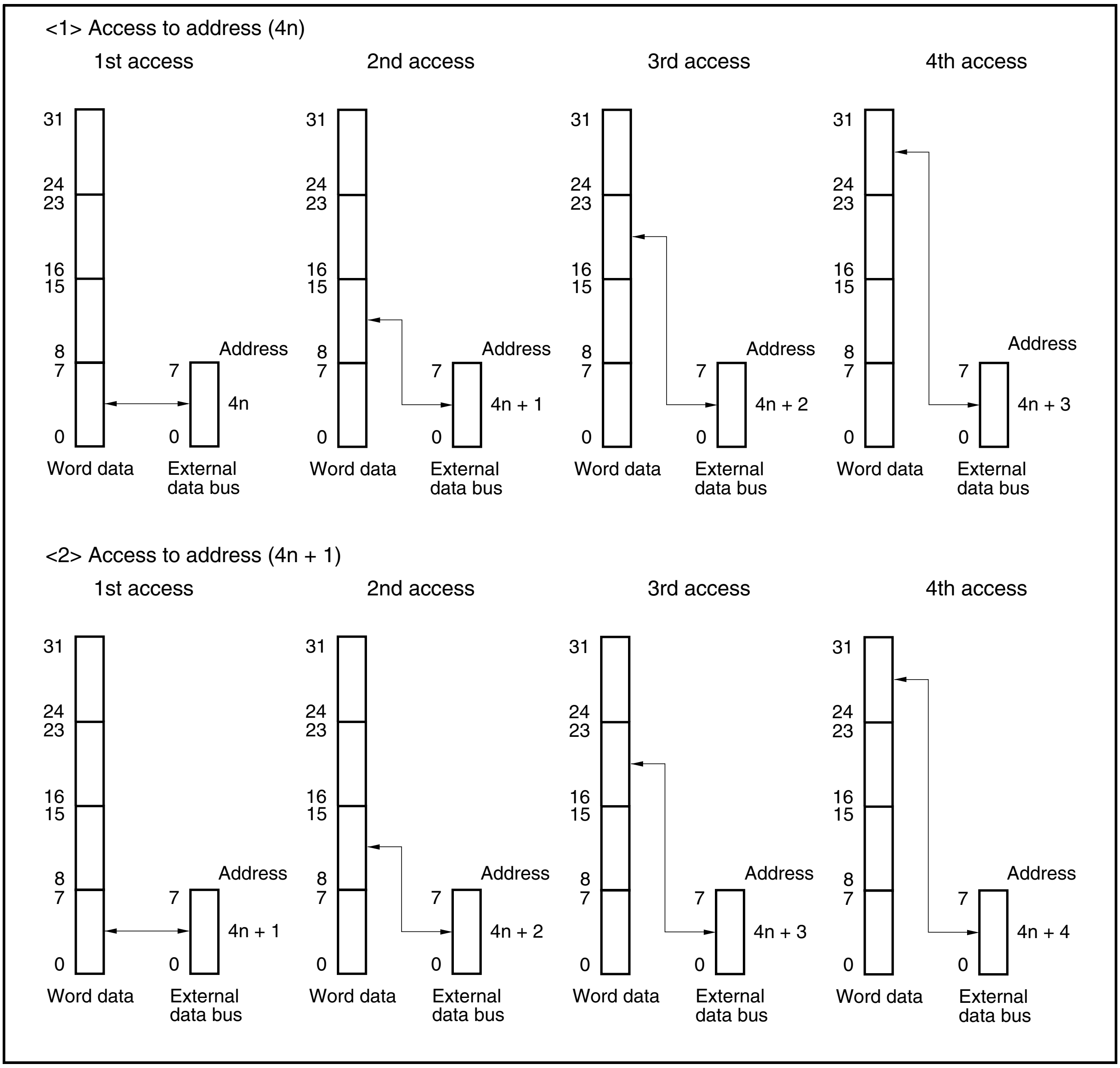


(a) When the data bus width is 16 bits (2/2)

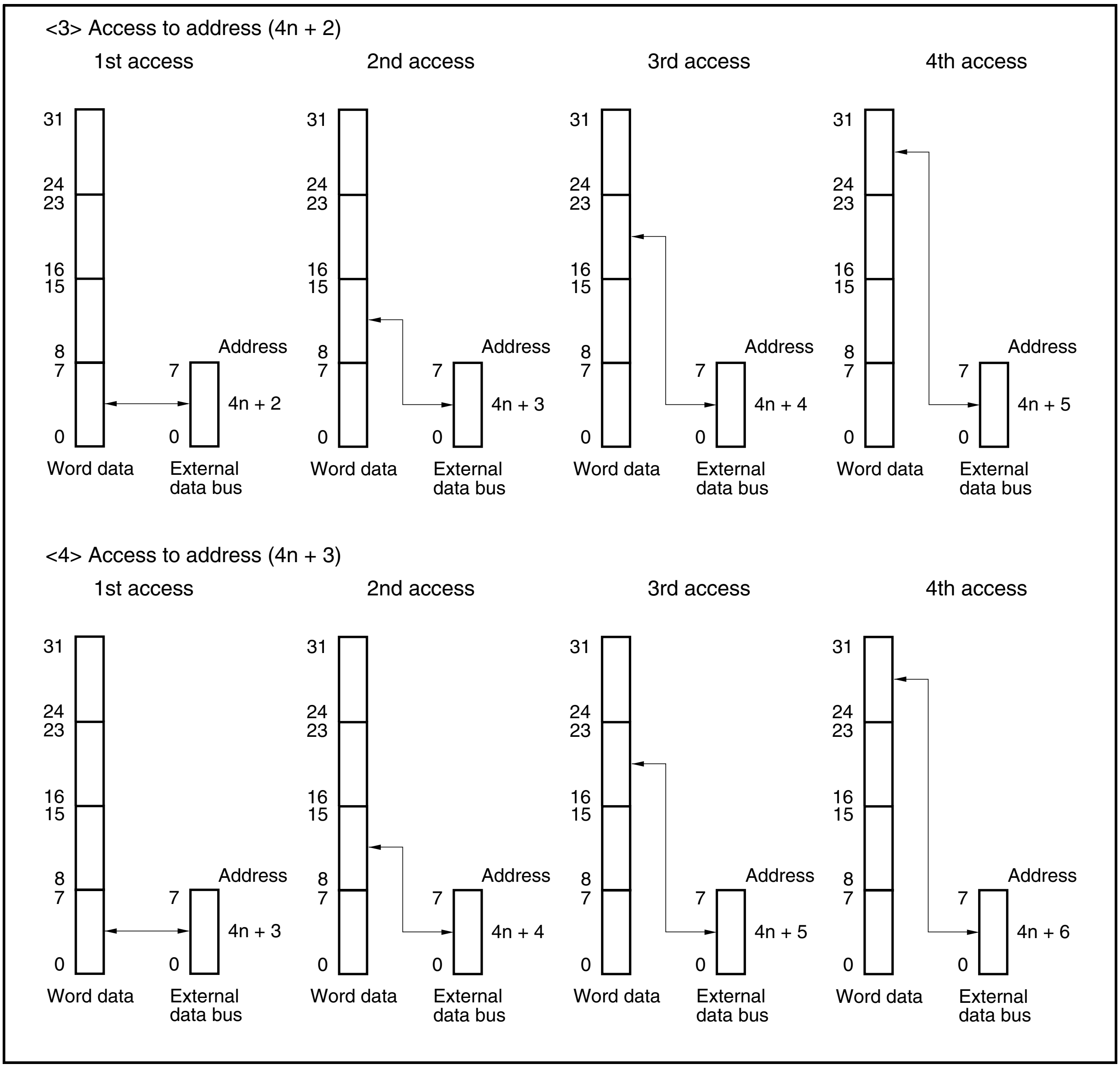


(b) When the data bus width is 8 bits (1/2)

32-bit data is transmitted/received via an 8-bit bus. Therefore, the data is transmitted/received in four accesses. The data is transmitted/received to/from the specified even/odd address of the external data bus.



(b) When the data bus width is 8 bits (2/2)



<3> Access to address (4n + 2)

<4> Access to address (4n + 3)

19.6 Wait Function

19.6.1 Programmable wait function

(1) Data wait control register 0 (DWC0)

To facilitate interfacing with a low-speed memory or I/O device and creating an interface circuit, it is possible to insert up to 7 data wait states in the starting bus cycle[Note] for each chip select space.

The number of wait states can be specified by program using the DWC0 register. Just after system reset, all blocks have 7 data wait states inserted.

This register can be read or written in 16-bit units.

Reset sets this register to 7777H.

Note SRAM read/write cycle

Cautions 1. The internal ROM and internal RAM areas are not subject to programmable waits and ordinarily no wait access is carried out.

The on-chip peripheral I/O area is not subject to programmable waits, with wait control performed by each on-chip peripheral function only.

2. Write to the DWC0 register after reset, and then do not change the set value. Also, when changing the initial values of the DWC0 register, do not access an external memory area until the settings are complete. However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 7777H R/W Address: FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	1	1	1	0	1[Note]	1[Note]	1
$\overline{\text{CSn}}$ signal								

	7	6	5	4	3	2	1	0
	0	DW12	DW11	DW10	0	DW02	DW01	DW00
$\overline{\text{CSn}}$ signal			$\overline{\text{CS1}}$				$\overline{\text{CS0}}$	

Note If the USB function controller (USBF) is used, it is recommended to set bits 9 and 10 to "0".
The other bits of the DWC0 register can be set simultaneously.

DWn2	DWn1	DWn0	Specification of number of wait states inserted in chip select space (n = 0, 1)
0	0	0	Not inserted
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Caution Be sure to set bits 3, 7, 11, and 15 to "0", and set bits 12 to 14 to "1". If they are set to values other than these, the operation is not guaranteed.

(2) Address wait control register (AWC)

This register is used to secure the setup and hold time for the address latch.

Address-setup or address-hold waits to be inserted in each bus cycle can be set by using the AWC register.

The address-setup wait state is inserted before the T1 state and the address-hold wait state is inserted after the T1 state.

Address-setup and address-hold wait state insertion can be set for each CS space.

This register can be read or written in 16-bit units.

Reset sets this register to FFFFH.

Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to address setup wait state and address hold wait state insertion.

2. During address setup wait state and address hold wait state, the $\overline{\text{WAIT}}$ pin-based external wait function is disabled.

3. Write the AWC register after reset, and then do not change the set values. Also, when changing the initial values of the AWC register, do not access an external memory area until the settings are complete.

After reset: FFFFH R/W Address: FFFFF488H

	15	14	13	12	11	10	9	8
AWC	1	1	1	1	1	1	1	1
$\overline{\text{CSn}}$ signal								

	7	6	5	4	3	2	1	0
	1	1	1[Note]	1[Note]	AHW1	ASW1	AHW0	ASW0
$\overline{\text{CSn}}$ signal					$\overline{\text{CS1}}$		$\overline{\text{CS0}}$	

Note If the USB function controller (USBF) is used, it is recommended to set bits 4 and 5 to "0".
The other bits of the AWC register can be set simultaneously.

AHWn	Specification of address hold wait state inserted in chip select space (n = 0, 1)
0	Not inserted
1	Inserted

ASWn	Specification of address setup wait state inserted in chip select space (n = 0, 1)
0	Not inserted
1	Inserted

Caution Be sure to set bits 6 to 15 to "1". If they are set to "0", the operation is not guaranteed.

19.6.2 External wait function

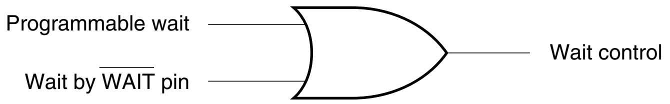
When a low-speed device or asynchronous system is connected, an arbitrary number of wait states can be inserted in the bus cycle by the external wait pin ($\overline{\text{WAIT}}$) for synchronization with the external device.

Just as with programmable waits, accessing internal ROM, internal RAM, and on-chip peripheral I/O areas cannot be controlled by external waits.

The external $\overline{\text{WAIT}}$ signal can be input asynchronously to the external bus clock frequency.

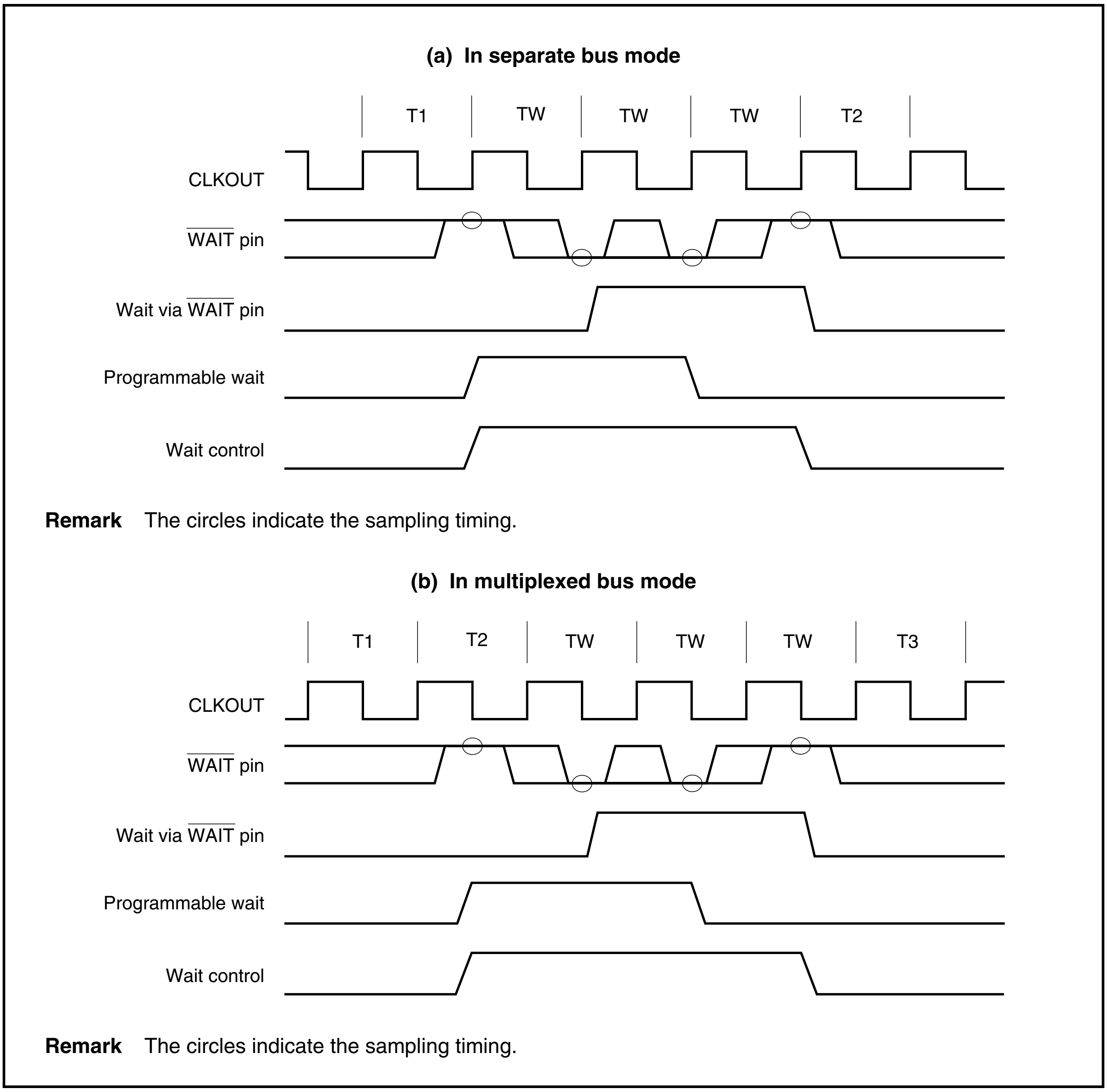
19.6.3 Relationship between programmable wait and external wait

A wait cycle is inserted as the result of an OR operation between the wait cycle specified by the set value of the programmable wait and the wait cycle controlled by the $\overline{\text{WAIT}}$ pin.



For example, if the timing of the programmable wait and the $\overline{\text{WAIT}}$ pin signal is as illustrated below, three wait states will be inserted in the bus cycle. Wait states inserted via the $\overline{\text{WAIT}}$ pin may be slower than the expected timing. In this case, use programmable waits to adjust the timing.

Figure 19-1. Example of Inserting Wait States



(a) In separate bus mode

T1
TW
TW
TW
T2
CLKOUT
$\overline{\text{WAIT}}$ pin
Wait via $\overline{\text{WAIT}}$ pin
Programmable wait
Wait control

Remark The circles indicate the sampling timing.

(b) In multiplexed bus mode

T1
T2
TW
TW
TW
T3
CLKOUT
$\overline{\text{WAIT}}$ pin
Wait via $\overline{\text{WAIT}}$ pin
Programmable wait
Wait control

Remark The circles indicate the sampling timing.

19.6.4 Bus cycles in which wait function is valid

In the V850E/IG4-H and V850E/IH4-H, the number of waits can be specified for each memory block. The following shows the bus cycles in which the wait function is valid and the registers used for wait setting.

Table 19-6. Bus Cycles in Which Wait Function Is Valid

Bus Cycle	Wait Type	Programmable Wait Setting			Wait by $\overline{\text{WAIT}}$ Pin
		Register	Bit	Number of Waits	
SRAM, external ROM, external I/O cycles	Address setup wait	AWC	ASWn	0, 1	× (invalid)
	Address hold wait	AWC	AHWn	0, 1	× (invalid)
	Data wait	DWC0	DWn2 to DWn0	0 to 7	√ (valid)

Remark n = 0, 1

19.7 Idle State Insertion Function

The idle state is inserted after a read cycle or a write cycle to the SRAM, external ROM, or external I/O.

(1) Bus cycle control register (BCC)

To facilitate interfacing with low-speed device devices, an idle state (TI) can be inserted into the current bus cycle after the T2 state (after TW state if a data wait state is inserted) to secure the data output float delay time on memory read access for chip select space. The bus cycle following the T2 state (or TW state) starts after the idle state is inserted.

An idle state can be inserted after a write access by using the bus clock division control register (DVC).

The idle state insertion setting can be specified by program using the BCC register. Immediately after the system reset, idle state insertion is automatically programmed for all memory blocks. For the timing when an idle state is inserted, see **19.8 Bus Timing**.

This register can be read or written in 16-bit units.

Reset sets this register to AAAAH.

Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to idle state insertion.

2. Write to the BCC register after reset, and then do not change the set values. Also, when changing the initial values of the BSC register, do not access an external memory area until the settings are complete. However, it is possible to access external memory areas whose initialization settings are complete.

3. The chip select signal ($\overline{\text{CS}}$n) does not become active in the idle state (n = 0, 1).

After reset: AAAAH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0
$\overline{\text{CSn}}$ signal								

	7	6	5	4	3	2	1	0
	1	0	1[Note]	0	BC11	0	BC01	0
$\overline{\text{CSn}}$ signal					$\overline{\text{CS1}}$		$\overline{\text{CS0}}$	

Note If the USB function controller (USBF) is used, it is recommended to set bit 5 to “0”. The other bits of the BCC register can be set simultaneously.

BCn1	Specification of idle state inserted in chip select space (n = 0, 1)
0	Not inserted
1	Inserted
Insertion of an idle state can be specified for chip select space after completion of a read cycle or a write cycle. If the DVC.BCWI bit = 0, however, the idle state is inserted only after completion of a read cycle and not after completion of a write cycle.	

Caution Be sure to set bits 0, 2, 4, 6, 8, 10, 12, and 14 to “0”, and set bits 7, 9, 11, 13, and 15 to “1”. If they are set other than above, the operation is not guaranteed.

(2) Bus clock division control register (DVC)

The DVC register is used to specify insertion of an idle state (TI) after completion of a write cycle.

This register can be read or written in 8-bit units.

Reset sets this register to 83H.

Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to idle state insertion.

2. Write to the DVC register after reset once (initial setting), and then do not change the set value. Also, do not access an external memory area until the initial setting of the DVC register is complete.
However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 83H R/W Address: FFFFF48EH

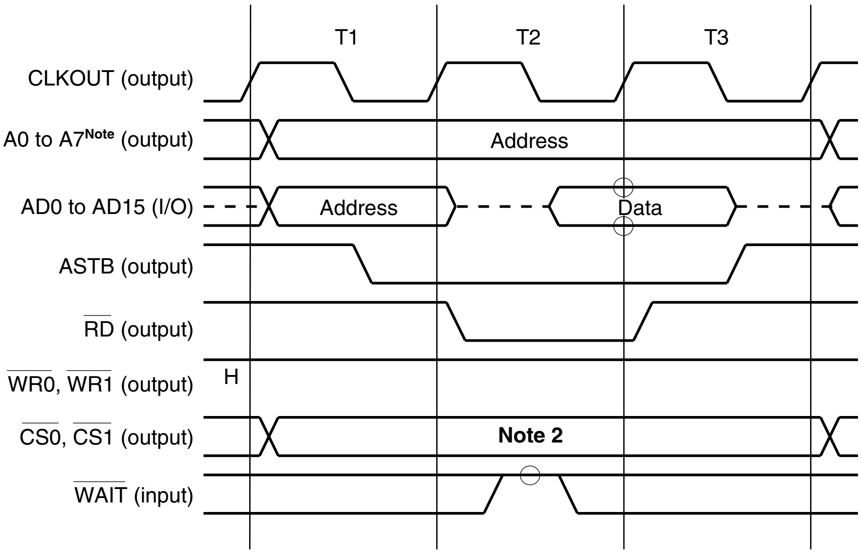
	7	6	5	4	3	2	1	0
DVC	BCWI	0	0	0	0	0	1	1

BCWI	Specification of idle state inserted after write cycle ends
0	Not inserted
1	Inserted (only when BCC.BC01 and BC11 bits = 1)

Caution Be sure to set bits 2 to 6 to “0”, bits 0 and 1 to “1”. If they are set to another value the operation is not guaranteed.

19.8 Bus Timing

(1) Read cycle (basic cycle)



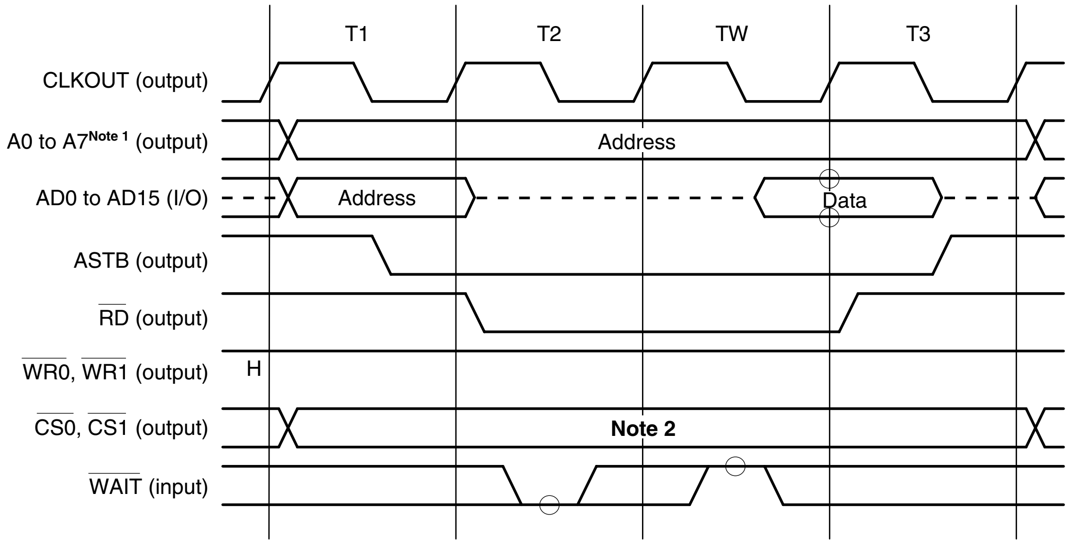
Notes 1. V850E/IH4-H only

2. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(2) Read cycle (when data wait state (1 wait) insertion)



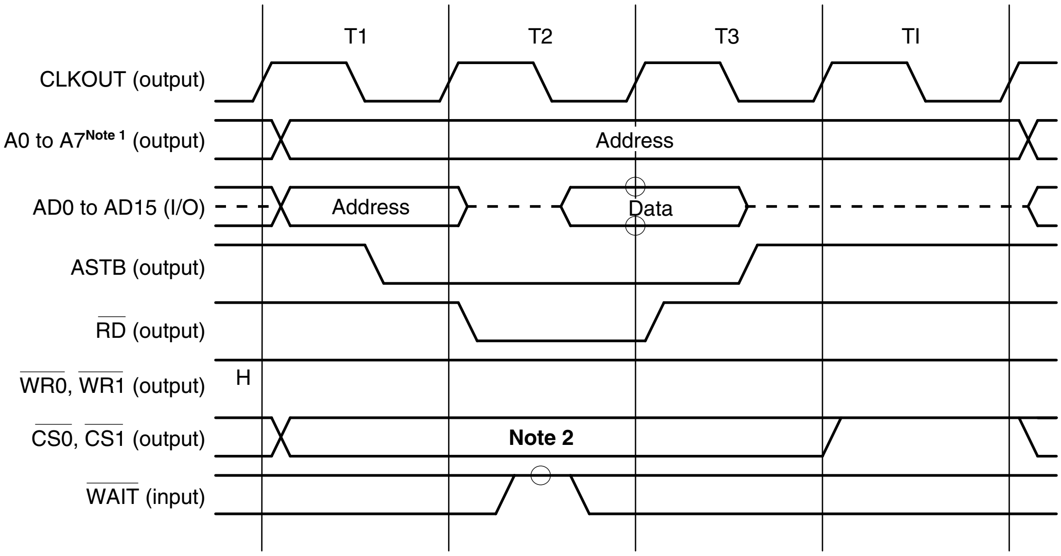
Notes 1. V850E/IH4-H only.

2. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(3) Read cycle (when idle state insertion)



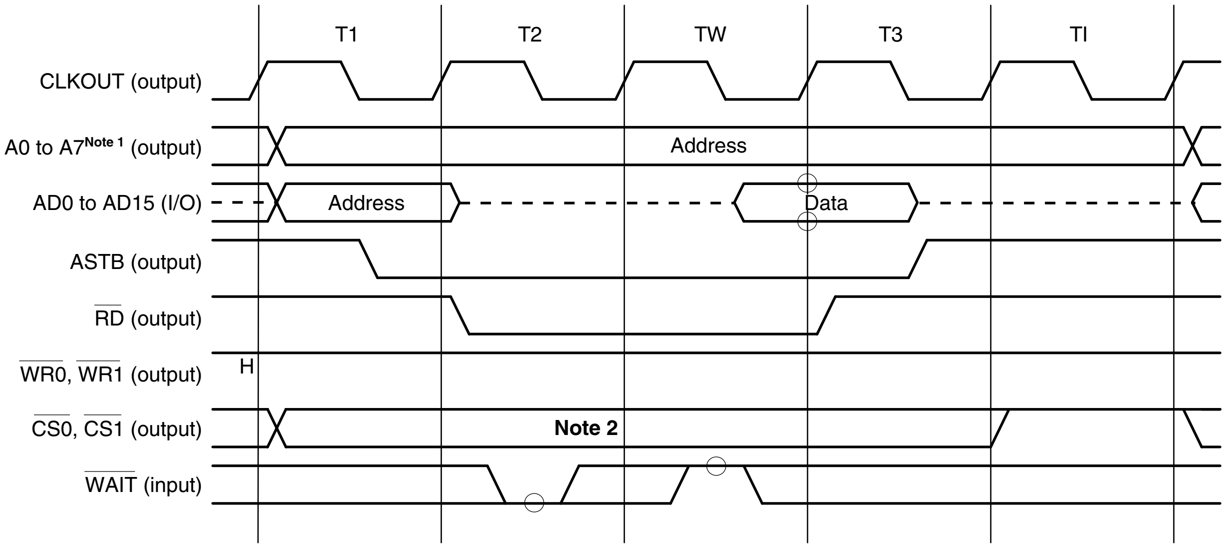
Notes 1. V850E/IH4-H only.

2. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(4) Read cycle (when data wait state (1 wait), idle state insertion)



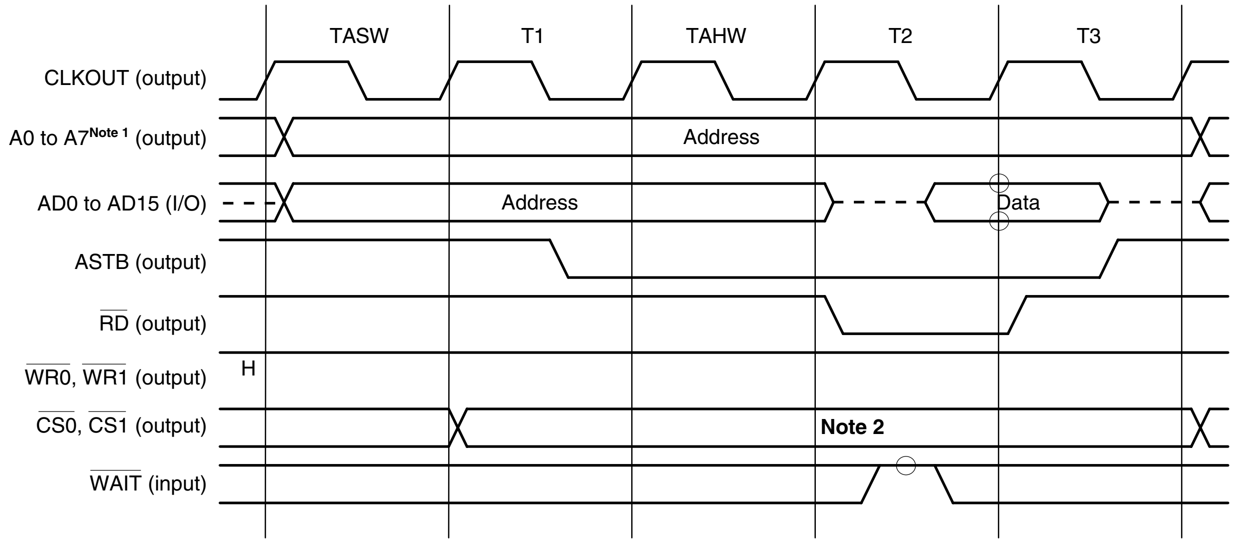
Notes 1. V850E/IH4-H only.

2. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(5) Read cycle (when address setup wait state, address hold wait state insertion)



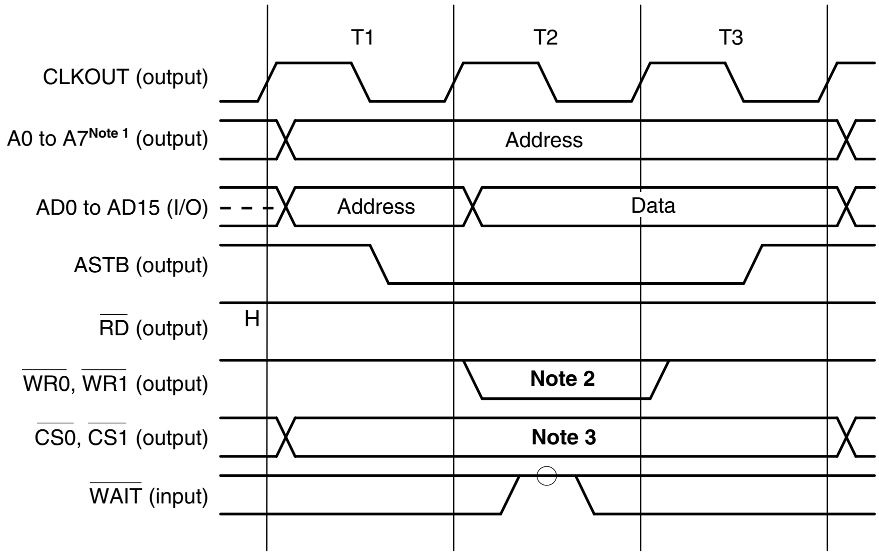
Notes 1. V850E/IH4-H only.

Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(6) Write cycle (basic cycle)



Notes 1. V850E/IH4-H only

2. The levels of these signals are as follows, depending on the access data bus width.

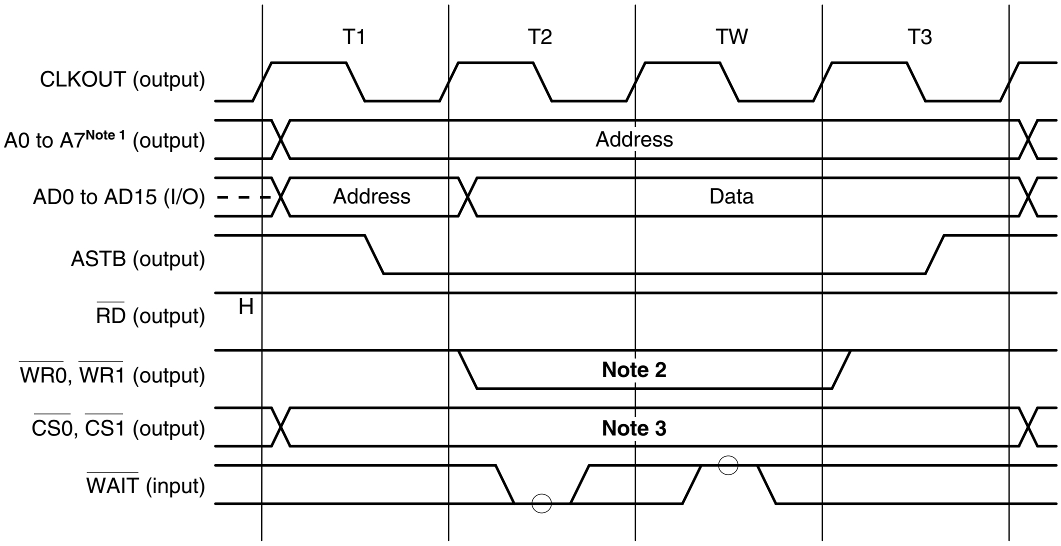
Access Data Bus Width	WR1	WR0
16 bits	Low level	Low level
8 bits	High level	Low level

3. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(7) Write cycle (when data wait state (1 wait) insertion)



Notes 1. V850E/IH4-H only

2. The levels of these signals are as follows, depending on the access data bus width.

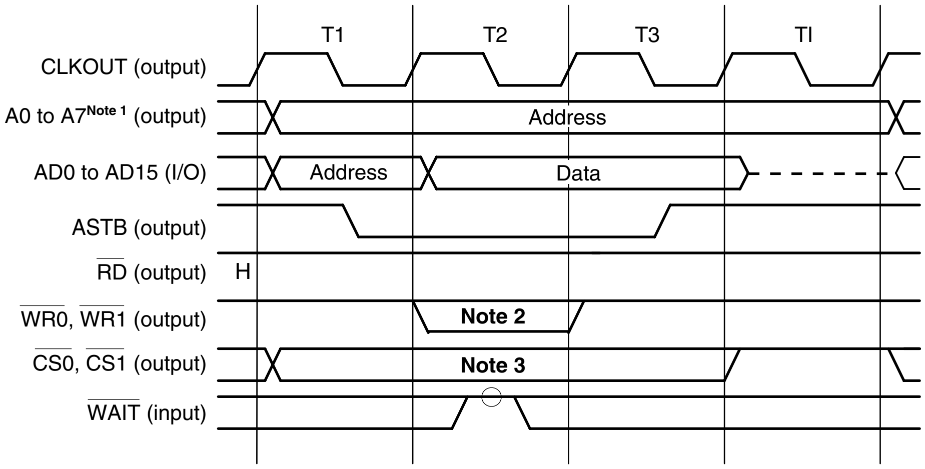
Access Data Bus Width	WR1	WR0
16 bits	Low level	Low level
8 bits	High level	Low level

3. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(8) Write cycle (when idle state insertion)



Notes 1. V850E/IH4-H only

2. The levels of these signals are as follows, depending on the access data bus width.

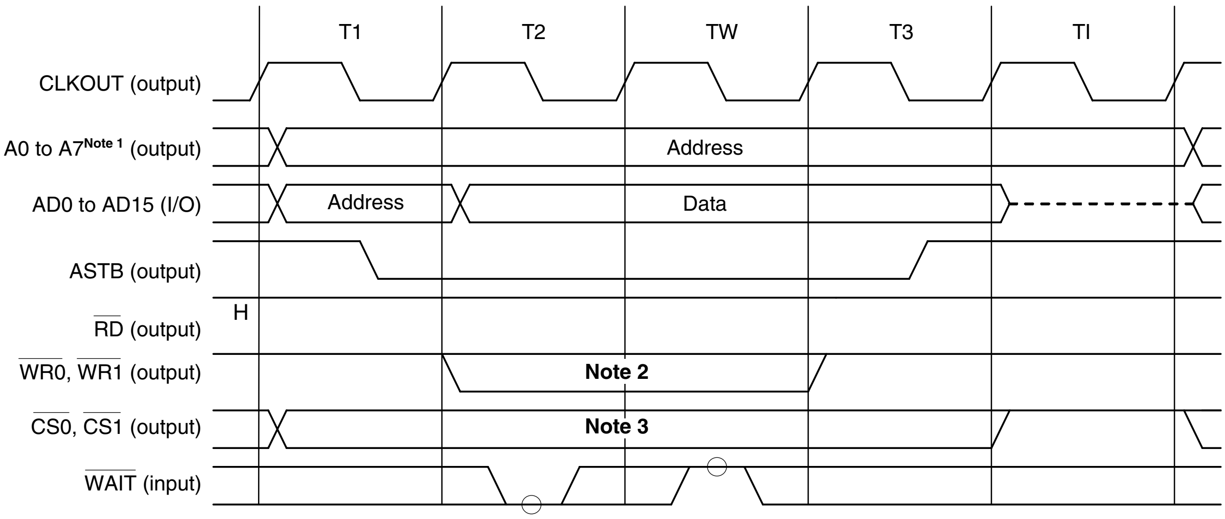
Access Data Bus Width	$\overline{\text{WR1}}$	$\overline{\text{WR0}}$
16 bits	Low level	Low level
8 bits	High level	Low level

3. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(9) Write cycle (when data wait state (1 wait), idle state insertion)



Notes 1. V850E/IH4-H only

2. The levels of these signals are as follows, depending on the access data bus width.

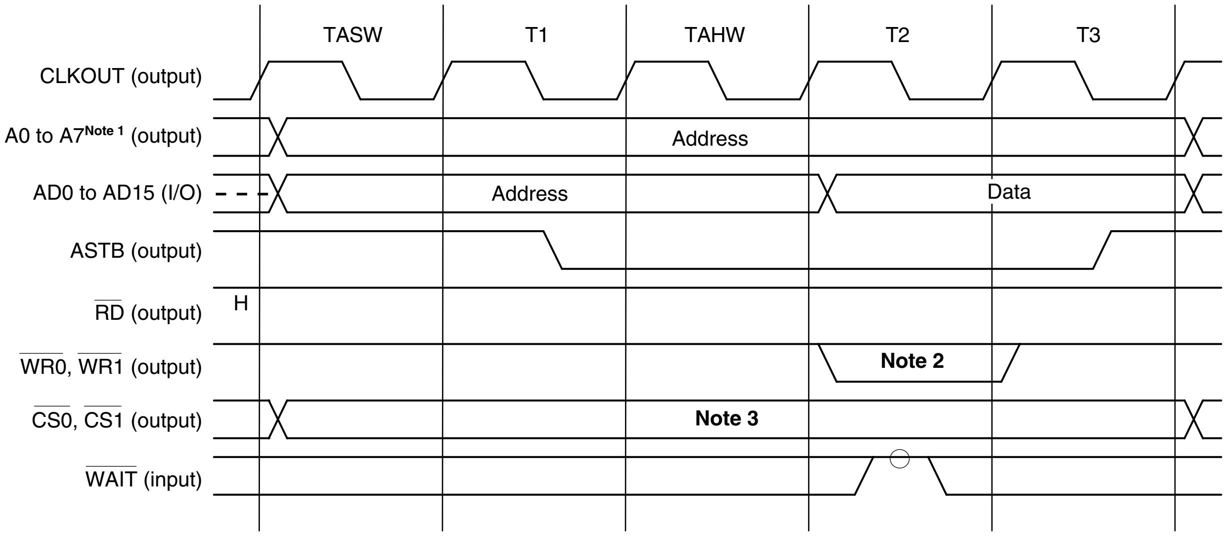
Access Data Bus Width	$\overline{WR1}$	$\overline{WR0}$
16 bits	Low level	Low level
8 bits	High level	Low level

3. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(10) Write cycle (when address setup wait state, address hold wait state insertion)



Notes 1. V850E/IH4-H only

2. The levels of these signals are as follows, depending on the access data bus width.

Access Data Bus Width	WR1	WR0
16 bits	Low level	Low level
8 bits	High level	Low level

3. Only the chip select spaces that can be accessed become active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

19.9 Bus Priority Order

There are two external bus cycles: instruction fetch and operand data access.

Operand data access has higher priority and instruction fetch has lower priority.

However, an instruction fetch may be inserted between a read access and write access during a read modify write access.

Table 19-7. Bus Priority Order

Priority Order	External Bus Cycle	Bus Master
High	Operand data access	CPU
Low	Instruction fetch	CPU

19.10 Boundary Operation Conditions

19.10.1 Program space

Branching to the on-chip peripheral I/O area is prohibited. If the above is performed, undefined data is fetched, and fetching from the external memory is not performed.

19.10.2 Data space

The V850E/IG4-H and V850E/IH4-H are provided with an address misalign function.

Through this function, data can be allocated to all addresses, regardless of the data format (word or halfword). In the case of word data and halfword data, however, the bus cycle will be generated at least twice if data is not aligned to the boundary, which causes the bus efficiency to drop.

(1) In the case of halfword-length data access

When the address's LSB is 1, a byte-length bus cycle will be generated 2 times.

(2) In the case of word-length data access

(a) When the address's LSB is 1, bus cycles will be generated in the order of byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle.

(b) When the address's lower 2 bits are 10, a halfword-length bus cycle will be generated 2 times.

CHAPTER 20 DMA (DMA CONTROLLER)

The V850E/IG4-H and V850E/IH4-H include a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfer between the internal RAM and on-chip peripheral I/O based on interrupt requests issued by the peripheral I/O (serial interface, timer, A/D converter, interrupts from an external input pin), or DMA transfer requests triggered by software.

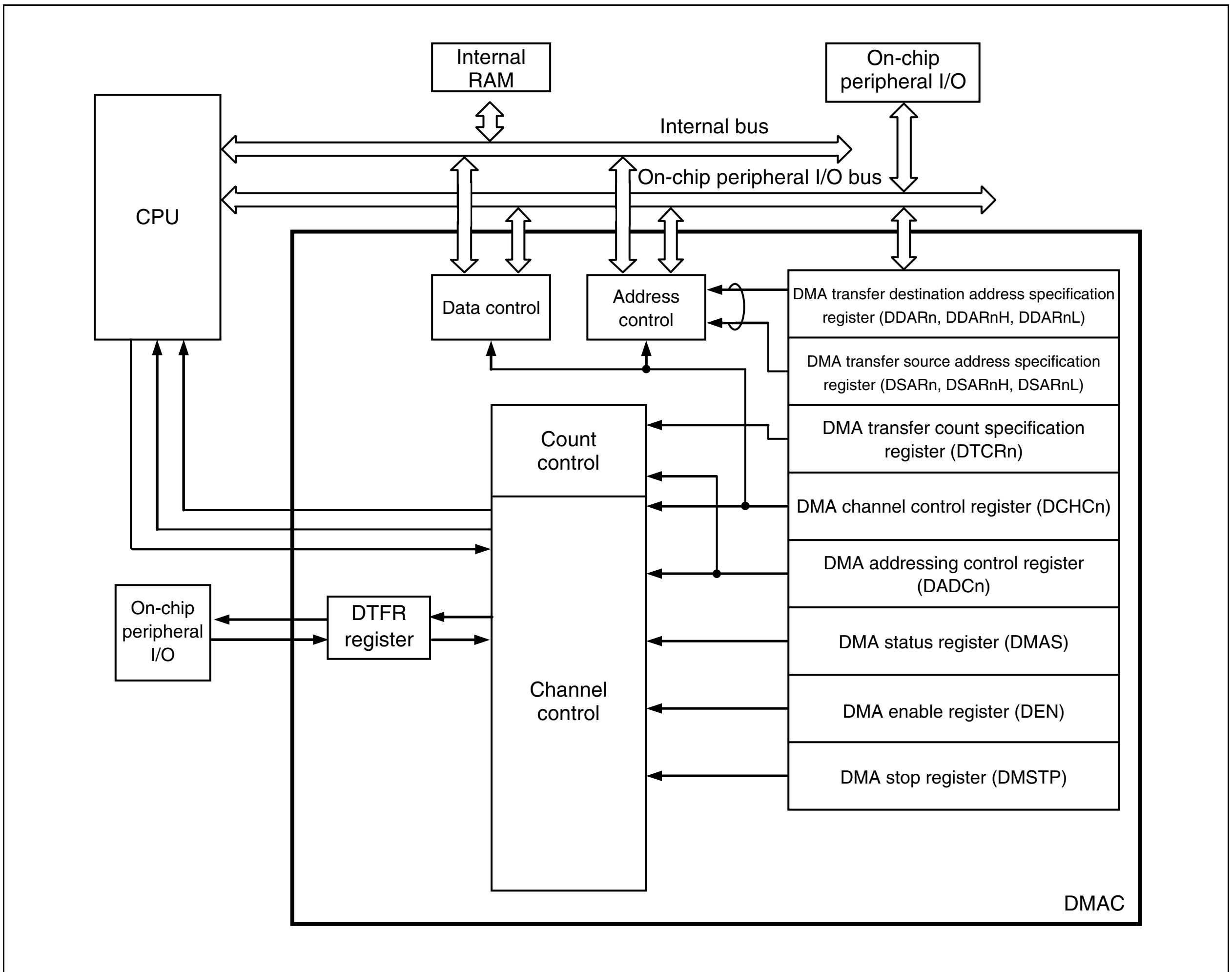
20.1 Features

- 7 independent DMA channels
- Transfer unit: 8/16/32 bits
- Maximum transfer count: 4096
- Transfer type: Two-cycle transfer
- Two transfer modes
  - Single transfer mode
  - Single-step transfer mode
- Transfer request
  - Request by interrupts from on-chip peripheral I/O (serial interface, timer, A/D converter) or interrupts from an external input pin
  - Requests triggered by software
- Transfer sources and destinations
  - Internal RAM ↔ on-chip peripheral I/O
- Next address setting function

20.2 Configuration

20.2.1 DMAC configuration

Figure 20-1. Block Diagram of DMAC


Internal RAM
On-chip peripheral I/O
Internal bus
On-chip peripheral I/O bus
CPU
Data control
Address control
DMA transfer destination address specification register (DDARn, DDARnH, DDARnL)
DMA transfer source address specification register (DSARn, DSARnH, DSARnL)
Count control
DMA transfer count specification register (DTCRn)
DMA channel control register (DCHCn)
DMA addressing control register (DADCn)
On-chip peripheral I/O
DTFR register
DMA status register (DMAS)
Channel control
DMA enable register (DEN)
DMA stop register (DMSTP)
DMAC

Remark n = 0 to 6

20.2.2 Operation outline

Channel n is activated by the start trigger selected by the DMA trigger factor register (DTFRn). After the start trigger is generated, DMA transfer is executed from the address (on-chip peripheral I/O or internal RAM address) specified by the DMA transfer source address specification register (DSARn, DSARnH, DSARnL) to the address (on-chip peripheral I/O or internal RAM address) specified by DMA transfer destination address specification register (DDARn, DDARnH, DDARnL).

When the number of DMA transfers specified by the DMA transfer count specification register (DTCRn) is completed, a DMA transfer end interrupt (INTDMAn) is generated.

Remark n = 0 to 6

20.2.3 Number of DMA transfer clock cycles

The number of DMA transfer clock cycles is as shown below.

Table 20-1. Minimum Number of DMA Transfer Clock Cycles

Transfer Target	Transfer Unit	Minimum Number of DMA Transfer Clock Cycles (f_{CLK})
Internal RAM → On-chip peripheral I/O	Byte/Halfword	8 clock cycles
	Word	14 clock cycles
On-chip peripheral I/O → Internal RAM	Byte/Halfword	6 clock cycles
	Word	12 clock cycles

Remarks 1. For the settings of the DMAWC0 and DMAWC1 registers, see **3.4.10 DMA wait control registers 0, 1 (DMAWC0, DMAWC1)**.

2. f_{CLK}: Internal system clock

20.3 Control Registers

20.3.1 DMA transfer destination address specification registers 0 to 6 (DDAR0 to DDAR6)

These registers are used to set the DMA transfer destination address (17 bits) for DMA channel n.

They are incremented at each DMA transfer based on the DADCn register setting.

Since these registers are 2-stage FIFO buffer registers that consist of a master register and a slave register, a new transfer destination address for DMA transfer can be specified during DMA transfer (see **20.9 Buffer Register Configuration**).

The value to be shown when these registers are read differs depending on the DCHCn.ENn bit or the DEN.ENnn bit.

ENn Bit or ENnn Bit[Note]	0	1
Value shown when register is read	Master register value	Slave register value

Note The ENn bit setting is applied to the ENnn bit, and the ENnn bit setting is applied to the ENn bit.

The DDARn register can be read or written in 32-bit units.

The DDARnH register is the higher 16 bits of the DDARn register and the DDARnL register is the lower 16 bits. These registers can be read or written in 16-bit units.

Reset makes these registers undefined.

Cautions 1. To accessing the DDARn registers in 32-bit units from the CPU, the result will be a misaligned access because the lower 2 bits of the address are not 00B. To access the DDARn registers when the program in the internal RAM is executed, be sure to access the registers in 16-bit units (see 20.14 (4) Program execution in internal RAM and DMA transfer).

2. The setting of the DARn0 bit is invalid during 16-bit transfer.

3. The settings of the DARn1 and DARn0 bits are invalid during 32-bit transfer.

4. When setting the DTCRn register and specifying an address using the DDARn register, do not specify an address that is in an address space to which neither the internal RAM nor an on-chip peripheral I/O is allocated; otherwise the operation is not guaranteed.

5. Set the DIRn bit after setting the DCHCn.ENn and DEN.ENnn bits to 0.

6. The value of the DIRn bit must not be the same as the value of the DSARn.SIRn bit.

Remarks 1. If 0 is written to a bit that is fixed to 1, the written value is ignored, and 1 is read.

2. If 1 is written to a bit that is fixed to 0, the written value is ignored, and 0 is read.

After reset: Undefined R/W Address: DDAR0 FFFFF086H,
DDAR0L FFFFF086H, DDAR0H FFFFF088H,
DDAR1 FFFFF096H,
DDAR1L FFFFF096H, DDAR1H FFFFF098H,
DDAR2 FFFFF0A6H,
DDAR2L FFFFF0A6H, DDAR2H FFFFF0A8H,
DDAR3 FFFFF0B6H,
DDAR3L FFFFF0B6H, DDAR3H FFFFF0B8H,
DDAR4 FFFFF0C6H,
DDAR4L FFFFF0C6H, DDAR4H FFFFF0C8H,
DDAR5 FFFFF0D6H,
DDAR5L FFFFF0D6H, DDAR5H FFFFF0D8H,
DDAR6 FFFFF0E6H,
DDAR6L FFFFF0E6H, DDAR6H FFFFF0E8H

	31	30	29	28	27	26	25	24
DDARn (DDARnH)	DIRn	0	0	0	1	1	1	1
(n = 0 to 6)	23	22	21	20	19	18	17	16
	1	1	1	1	1	1	1	DARn16
	15	14	13	12	11	10	9	8
(DDARnL)	DARn15	DARn14	DARn13	DARn12	DARn11	DARn10	DARn9	DARn8
	7	6	5	4	3	2	1	0
	DARn7	DARn6	DARn5	DARn4	DARn3	DARn2	DARn1	DARn0

DIRn	DMA transfer destination specification
0	On-chip peripheral I/O
1	Internal RAM

DARn16 to DARn0	Sets the DMA transfer destination address (A16 to A0). When the DCHCn.ENn bit or the DEN.ENnn bit is set to 1 by the DADCn register, the register values change after each DMA transfer. For details, see Table **20-2 DDARn Register Values Set in Accordance with DADCn Register Setting**.

Table 20-2. DDARn Register Values Set in Accordance with DADCn Register Setting

DADCn Register				DDARn Register Value Increase/Decrease
DSn1 Bit	DSn0 Bit	DADn1 Bit	DADn0 Bit	
0	0	0	0	+1
		0	1	–1
		1	0	0
0	1	0	0	+2
		0	1	–2
		1	0	0
1	0	0	0	+4
		0	1	–4
		1	0	0
Other than above				Setting prohibited

20.3.2 DMA transfer source address specification registers 0 to 6 (DSAR0 to DSAR6)

These registers are used to set the DMA transfer source address (17 bits) for DMA channel n.

They are incremented at each DMA transfer based on the DADCn register setting.

Since these registers are 2-stage FIFO buffer registers that consist of a master register and a slave register, a new source address for DMA transfer can be specified during DMA transfer (see **20.9 Buffer Register Configuration**).

The value to be shown when these registers are read differs depending on the DCHCn.ENn bit or the DEN.ENnn bit.

ENn Bit or ENnn Bit[Note]	0	1
Value shown when register is read	Master register value	Slave register value

Note The ENn bit setting is applied to the ENnn bit. The ENnn bit setting is applied to the ENn bit.

The DSARn register can be read or written in 32-bit units.

The DSARnH register is the higher 16 bits of the DSARn register and the DSARnL register is the lower 16 bits. These registers can be read or written in 16-bit units.

Reset makes these registers undefined.

Cautions 1. When accessing these registers in 32-bit units, the result will be a misaligned access because the lower 2 bits of the address are not 00B. To access the DSARn registers when the program in the internal RAM is executed, be sure to access these registers in 16-bit units (see 20.14 (4) Program execution in internal RAM and DMA transfer).

2. The setting of the SARn0 bit is invalid during 16-bit transfer.

3. The settings of the SARn1 and SARn0 bits are invalid during 32-bit transfer.

4. When setting the DTCRn register and specifying an address using the DSARn register, do not specify an address that is in an address space to which neither the internal RAM nor an on-chip peripheral I/O is allocated; otherwise the operation is not guaranteed.

5. Set the SIRn bit after setting the DCHCn.ENn and DEN.ENnn bits to 0.

6. The value of the SIRn bit must not be the same as the value of the DSARn.DIRn bit.

Remarks 1. If 0 is written to a bit that is fixed to 1, the written value is ignored, and 1 is read.

2. If 1 is written to a bit that is fixed to 0, the written value is ignored, and 0 is read.

After reset: Undefined R/W Address: DSAR0 FFFFF08AH,
DSAR0L FFFFF08AH, DSAR0H FFFFF08CH,
DSAR1 FFFFF09AH,
DSAR1L FFFFF09AH, DSAR1H FFFFF09CH,
DSAR2 FFFFF0AAH,
DSAR2L FFFFF0AAH, DSAR2H FFFFF0ACH,
DSAR3 FFFFF0BAH,
DSAR3L FFFFF0BAH, DSAR3H FFFFF0BCH,
DSAR4 FFFFF0CAH,
DSAR4L FFFFF0CAH, DSAR4H FFFFF0CCH,
DSAR5 FFFFF0DAH,
DSAR5L FFFFF0DAH, DSAR5H FFFFF0DCH,
DSAR6 FFFFF0EAH,
DSAR6L FFFFF0EAH, DSAR6H FFFFF0ECH

	31	30	29	28	27	26	25	24
DSARn (DSARnH)	SIRn	0	0	0	1	1	1	1
(n = 0 to 6)	23	22	21	20	19	18	17	16
	1	1	1	1	1	1	1	SARn16

	15	14	13	12	11	10	9	8
(DSARnL)	SARn15	SARn14	SARn13	SARn12	SARn11	SARn10	SARn9	SARn8
	7	6	5	4	3	2	1	0
	SARn7	SARn6	SARn5	SARn4	SARn3	SARn2	SARn1	SARn0

SIRn	DMA transfer source specification
0	On-chip peripheral I/O
1	Internal RAM

SARn16 to SARn0	Sets the DMA transfer source address (A16 to A0). When the DCHCn.ENn bit or the DEN.ENnn bit is set to 1 by the DADCn register, the register values change after each DMA transfer. For details, see **Table 20-3 DSARn Register Values Set in Accordance with DADCn Register Setting**.

Table 20-3. DSARn Register Values Set in Accordance with DADCn Register Setting

DADCn Register				DSARn Register Value Increase/Decrease
DSn1 Bit	DSn0 Bit	SADn1 Bit	SADn0 Bit	
0	0	0	0	+1
		0	1	−1
		1	0	0
0	1	0	0	+2
		0	1	−2
		1	0	0
1	0	0	0	+4
		0	1	−4
		1	0	0
Other than above				Setting prohibited

20.3.3 DMA transfer count specification registers 0 to 6 (DTCR0 to DTCR6)

These 16-bit registers are used to set the number of transfers for DMA channel n. These registers store the remaining number of transfers during DMA transfer.

Since these registers are 2-stage FIFO buffer registers that consist of a master register and a slave register, a new transfer count for DMA transfer can be specified during DMA transfer (see **20.9 Buffer Register Configuration**).

The value to be shown when these registers are read differs depending on the DCHCn.ENn bit or the DEN.ENnn bit.

ENn Bit or ENnn Bit[Note]	0	1
Value shown when register is read	Master register value	Slave register value

Note The ENn bit setting is applied to the ENnn bit. The ENnn bit setting is applied to the ENn bit.

These registers can be read or written in 16-bit units.
Reset makes these registers undefined.

Cautions 1. When the next address function is not used (DCHCn.MLEn bit = 0), the value of the DTCRn register is decremented at each DMA transfer, and the set value of the DTCRn register is restored when DMA transfer ends.

2. When the next address function is used (DCHCn.MLEn bit = 1), the DTCRn register value is decremented at each DMA transfer, and the latest value written to the DTCRn register is reloaded when DMA transfer ends.

After reset: Undefined R/W Address: DTCR0 FFFFF084H, DTCR1 FFFFF094H,
DTCR2 FFFFF0A4H, DTCR3 FFFFF0B4H,
DTCR4 FFFFF0C4H, DTCR5 FFFFF0D4H,
DTCR6 FFFFF0E4H

DTCRn
(n = 0 to 6)

15	14	13	12	11	10	9	8
0	0	0	0	DTCRn11	DTCRn10	DTCRn9	DTCRn8

7	6	5	4	3	2	1	0
DTCRn7	DTCRn6	DTCRn5	DTCRn4	DTCRn3	DTCRn2	DTCRn1	DTCRn0

DTCRn11 to	Transfer count setting (number of remaining transfers retained during DMA transfer)	
DTCRn0	When writing	When reading
0000H	4096 transfers	Number of remaining transfers
0001H	1 transfer	
:	:	
0FFFH	4095 transfers	

20.3.4 DMA addressing control registers 0 to 6 (DADC0 to DADC6)

These 16-bit registers are used to specify the DMA transfer mode for DMA channel n. Do not change the DADCn register setting during the period from the start of DMA transfer to the end of the specified number of DMA transfers. If the DADCn register setting is changed, the operation is not guaranteed.

These registers can be read or written in 16-bit units.

Reset sets these registers to 0000H.

After reset: 0000H R/W Address: DADC0 FFFFF082H, DADC1 FFFFF092H,
DADC2 FFFFF0A2H, DADC3 FFFFF0B2H,
DADC4 FFFFF0C2H, DADC5 FFFFF0D2H,
DADC6 FFFFF0E2H

	15	14	13	12	11	10	9	8
DADCn	0	0	0	0	0	0	0	0
(n = 0 to 6)	7	6	5	4	3	2	1	0
	DSn1	DSn0	SADn1	SADn0	DADn1	DADn0	TMn	0

DSn1	DSn0	Setting of the transfer data size for DMA channel n
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	Setting prohibited

SADn1	SADn0	Setting of the count direction of the transfer source address for DMA channel n
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

DADn1	DADn0	Setting of the count direction of the transfer destination address for DMA channel n
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

TMn	Setting of DMA transfer mode
0	Single transfer mode
1	Single-step transfer mode

20.3.5 DMA channel control registers 0 to 6 (DCHC0 to DCHC6)

These 16-bit registers are used to specify the DMA transfer operation mode for DMA channel n.

These registers can be read or written in 16-bit units. However, bit 7 is read-only.

Reset sets these registers to 0000H.

Cautions 1. If transfer ends with the MLEn bit set to 1, and the next transfer request is executed as a DMA transfer (hardware DMA) started by an interrupt from an on-chip peripheral I/O, the next transfer will be executed with the TCn bit set to 1 (this bit is not automatically cleared to 0).

2. If the ENn bit or DEN.ENnn bit is cleared to 0 while DMA transfer is stopped (by setting the STPn bit or DMSTP.STPnn bit to 1), DMA transfer cannot be resumed.

3. If the DMA transfer mode or DMA transfer start trigger is changed while DMA transfer is stopped (by setting the STPn bit or DMSTP.STPnn bit to 1), the operation after DMA is resumed (by setting the STPn or STPnn bit to 0) cannot be guaranteed.

4. In the two-stage FIFO type buffer registers (DDARn, DDARnL, DDARnH, DSARn, DSARnL, DSARnH, and DTCRn) data is transferred from the master register to the slave register if 1 is written to the ENn or DEN.ENnn bit while the bit is 0. If 1 is written to the ENn or ENnn bit while the bit is 1, the data is not transferred from the master register to the slave register.

5. When setting the STGn bit after the specified number of DMA transfer cycles is complete (indicated by TCn bit = 1 or DMAS.TCnn bit = 1) while the MLEn bit is 0, be sure to set the STGn bit to 1 after clearing the TCn or TCnn bit to 0 and then setting the ENn or ENnn bit to 1. If the STGn bit is set to 1 while the TCn or TCnn bit is 1, the setting of the STGn bit is ignored.

6. When setting the STGn bit after the specified number of DMA transfer cycles is complete (indicated by TCn bit = 1 or DMAS.TCnn bit = 1) while the MLEn bit is 1, be sure to set the STGn bit to 1 after clearing the TCn or TCnn bit to 0. If the STGn bit is set to 1 while the TCn or TCnn bit is 1, the setting of the STGn bit is ignored.

7. The ENn bit can only be changed from 0 to 1 while the TCn or TCnn bit is 0. Setting the ENn bit to 1 while the TCn or TCnn bit is 1 is ignored.

8. If the ENn bit is changed from 1 to 0 before the specified number of DMA transfer cycles are complete, DMA transfer will be terminated at the end of the current DMA transfer cycle. At this time, the TCn or TCnn bit will not be set to 1. However, if DMA transfer is terminated at the end of the final DMA transfer cycle meaning that the specified number of DMA transfer cycles are complete, the TCn or TCnn bit will be set to 1 and the DMA transfer end interrupt (INTDMAn) will be generated.

(1/2)

After reset: 0000H R/W Address: DCHC0 FFFFF08EH, DCHC1 FFFFF09EH,
DCHC2 FFFFF0AEH, DCHC3 FFFFF0BEH,
DCHC4 FFFFF0CEH, DCHC5 FFFFF0DEH,
DCHC6 FFFFF0EEH

	15	14	13	12	11	10	9	8
DCHCn	0	0	0	0	0	0	0	0
(n = 0 to 6)	7	6	5	4	3	2	1	0
	TCn	0	0	0	MLEn	STPn	STGn	ENn

TCn[Note]	Status flag indicating whether or not DMA transfer of DMA channel n has ended
0	DMA transfer had not ended.
1	DMA transfer had ended.

- This bit is set to 1 when DMA transfer ends and cleared to 0 when it is read.
- The written value is ignored even if writing to the TCn bit.
- Only “0” can be written to the DMAS.TCnn bit. Writing “0” to the TCnn bit can clear to 0 the corresponding TCn bit.
- The TCn bit setting is applied to the DMAS.TCnn bit.

MLEn	Next address setting function enable/disable specification
0	Next address setting function disabled
1	Next address setting function enabled

- If a terminal count occurs (if transfer has been completed the specified number of times) when the next address setting function is disabled (MLEn bit = 0), the ENn bit or the DEN.ENnn bit is cleared to 0 and DMA transfer is disabled. When the next DMA transfer is requested, the TCn bit or the DMAS.TCnn bit must be cleared to 0 and then the ENn or ENnn bit must be set to 1.
- If a terminal count occurs (if transfer has been completed the specified number of times) when the next address setting function is enabled (MLEn bit = 1), the ENn/ENnn bit is not cleared to 0 and DMA transfer remains enabled. When the next DMA transfer is an interrupt from an on-chip peripheral I/O (hardware DMA), the DMA transfer request is acknowledged even if the TCn/TCnn bit is not cleared to 0.
- The MLEn bit is enabled when DMA transfer is started by an interrupt source from an on-chip peripheral I/O. To trigger DMA by software (STGn bit = 1), clear the TCn/TCnn bit to 0 and set the STGn bit to 1.

Note The TCn bit is a read-only bit.

(2/2)

<table>
<tr><td>STPn</td><td>When the STPn bit is set to 1, DMA transfer is suspended. To restart DMA transfer, clear this bit to 0.
If the ENn bit or DEN.ENnn bit is cleared to 0 when this bit is set to 1, DMA transfer is forcibly terminated. DMA transfer cannot be restarted even if the STPn bit is cleared to 0 after setting the ENn/ENnn bit to 1 again (forcible termination).
In the single transfer mode, only one DMA transfer start trigger can be pended while DMA transfer is suspended by setting the STPn bit to 1 (even if there are two or more start triggers, only one is counted). The transfer is restarted after the STPn bit becomes 0. In the single-step transfer mode, DMA transfer start triggers are not pended and are ignored while DMA transfer is suspended by setting the STPn bit to 1.</td></tr>
<tr><td colspan="2">• The STPn bit is not cleared even when the ENn or ENnn bit is cleared to “0”.
• The setting of the STPn bit is applied setting the DMSTP.STPnn bit.</td></tr>
</table>

<table>
<tr><td>STGn</td><td>If the STGn bit is set to 1 in the DMA transfer enabled state (TCn bit or DMAS.TCnn bit = 0, ENn bit or DEN.ENnn bit = 1), DMA transfer is triggered by software.
This bit is always read as “0” and writing “0” to this bit is ignored.</td></tr>
<tr><td colspan="2">• Set the STGn bit to 1 after setting the ENn/ENnn bit to 1.
Writing 1 to the STGn bit is ignored when the ENn/ENnn bit = 0.
• Writing 1 to the STGn bit is ignored during the DMA transfer cycle in the single-transfer mode.
• Writing 1 to the STGn bit is ignored during the specified number of cycles of DMA transfer in the single-step transfer mode.</td></tr>
</table>

<table>
<tr><th>ENn</th><th>DMA transfer enable/disable specification for DMA channel n</th></tr>
<tr><td>0</td><td>DMA transfer disabled (forcibly terminated)</td></tr>
<tr><td>1</td><td>DMA transfer enabled</td></tr>
<tr><td colspan="2">• The ENn bit is cleared to 0 when the MLEn bit = 0 and DMA transfer ends.
• The ENn bit setting is applied to the DEN.ENnn bit.</td></tr>
</table>

20.3.6 DMA status register (DMAS)

The register shows the status of DMA channel n transfer. This register always corresponds to the DCHCn.TCn bit, but it can be cleared to 0 by writing 0 to it.

This register can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

After reset: 00H R/W Address: FFFFF0F0H

	7	6	5	4	3	2	1	0
DMAS	0	TC66	TC55	TC44	TC33	TC22	TC11	TC00

TCnn	Status flags indicating whether or not DMA transfer for DMA channel n has ended
0	DMA transfer has not ended
1	DMA transfer has ended

- These flags are set to 1 when specified number of times of DMA transfer ends.
- The TCnn bit cannot be cleared to 0 even if read.
- Only 0 can be written to the TCnn bit. Writing 0 to the TCnn bit will clear to 0 the corresponding DCHCn.TCn bit. Writing “1” to the TCnn bit is ignored.
- The TCnn bit setting is applied to the DCHCn.TCn bit.

Remark n = 0 to 6

20.3.7 DMA enable register (DEN)

This register enables and disables DMA transfer. The ENnn bit of this register is the same as the DCHCn.ENn bit. DMA transfer can be forcibly terminated by clearing the ENnn bit of the DMA channel n to 0.

This register can be read or written in 8-bit units.

Reset sets these registers to the 00H.

Cautions 1. DMA transfer cannot be restarted if the ENnn bit and the DCHC.ENn bit are cleared to 0 while DMA transfer is suspended (by setting the DCHCn.STPn bit or the DMSTP.STPnn bit to 1).

2. In the two-stage FIFO type buffer registers (DDARn, DDARnL, DDARnH, DSARn, DSARnL, DSARnH, and DTCRn) data is transferred from the master register to the slave register if 1 is written to the ENn or DEN.ENnn bit while the bit is 0. If 1 is written to the ENn or ENnn bit while the bit is 1, the data is not transferred from the master register to the slave register.

3. The ENn bit can only be changed from 0 to 1 while the TCn or TCnn bit is 0. Setting the ENnn bit to 1 while the TCn or TCnn bit is 1 is ignored.

4. If the ENnn bit is changed from 1 to 0 before the specified number of DMA transfer cycles are complete, DMA transfer will be terminated at the end of the current DMA transfer cycle. At this time, the TCn or TCnn bit will not be set to 1. However, if DMA transfer is terminated at the end of the final DMA transfer cycle meaning that the specified number of DMA transfer cycles are complete, the TCn or TCnn bit will be set to 1 and the DMA transfer end interrupt (INTDMAn) will be generated.

After reset: 00H R/W Address: FFFFF0F2H

	7	6	5	4	3	2	1	0
DEN	0	EN66	EN55	EN44	EN33	EN22	EN11	EN00

ENnn	DMA transfer enable/disable specification for DMA channel n
0	DMA transfer disabled (forcibly terminated)
1	DMA transfer enabled
The ENnn bit setting is applied to the DCHCn.ENn bit.	

Remark n = 0 to 6

20.3.8 DMA stop register (DMSTP)

This register suspends DMA transfer.

The STPnn bit of this register is the same as the DCHCn.STPn bit. DMA transfer can be restarted by clearing the STPnn bit of DMA channel n to 0.

This register can be read or written in 8-bit units.

Reset sets these registers to 00H.

Caution If the DMA transfer mode or DMA transfer start trigger is changed while DMA transfer is suspended (by setting the STPnn bit or DMSTP.STPn bit to 1), the operation after DMA is resumed (by setting the STPn or STPnn bit to 0) cannot be guaranteed.

After reset: 00H R/W Address: FFFFF0F4H

	7	6	5	4	3	2	1	0
DMSTP	0	STP66	STP55	STP44	STP33	STP22	STP11	STP00

STPnn	DMA transfer is suspended by setting the STPnn bit to 1. Clear this bit to 0 to restart DMA transfer. If the DCHC.ENn bit or DEN.ENnn bit is cleared to 0 when this bit is set to 1, DMA transfer is forcibly terminated. DMA transfer cannot be restarted even if the STPnn bit is cleared to 0 after setting the ENn/ENnn bit to 1 again (forcible termination). In the single transfer mode, only one DMA transfer start trigger can be pended while DMA transfer is suspended by settings the STPnn bit to 1 (even if there are two or more start triggers, only one is counted). The transfer is restarted after the STPnn bit becomes 0. In the single-step transfer mode, DMA transfer start triggers while DMA transfer is suspended by setting the STPnn bit to 1 are not pended and are ignored.
• The STPnn bit is not cleared even when the ENn/ENnn bit is cleared to “0”. • The setting of the STPnn bit is applied to the DCHCn.STPn bit.	

Remark n = 0 to 6

20.3.9 DMA trigger factor register n (DTFRn)

This 16-bit register is used to control DMA transfer start triggers generated by interrupt requests from on-chip peripheral I/O.

The interrupt requests set by using this registers serve as DMA transfer start triggers.

The DTFRn register can be read or written in 16-bit units.

The DTFRnH register is the higher 8 bits of the DTFRn register and the DTFRnL register is the lower 8 bits of the DTFRn register. These registers can be read and written in 8-bit units.

Reset sets these registers to 0000H.

Cautions 1. To change the IFCn0 to IFCn6 bits of the DTFRn register (except for clearing the DFn/DFm bit), disable DMA channel n for which the IFCn0 to IFCn6 bits are to be changed, and all other DMA channels having a priority lower than channel n (clear the DCHC.ENn bit or the DEN.ENnn bit to 0 and the ENm/ENmm bit to 0). To enable the DMA operation after changing the IFCn0 to IFCn6 bits of the DTFRn register (ENn/ENnn/ENm/ENmm bit = 1), be sure to clear the DFn/DFm bit.

Unless these conditions are satisfied, DMA channel m may perform the following operation.

- **DMAm transfer is started even when a DMAm start trigger is not generated.**
- **DMAm transfer is not started even when a DMAm start trigger is generated.**

2. To overwrite the same value to the DTFRn register, disable the operation of DMA channel n corresponding to the DTFRn register to which the same value is to be written (ENn/ENnn bit = 0). The operation of DMA channel m that has a priority lower than DMA channel n does not have to be disabled.

3. If data is written to the DTFRn register, a DMAn start request that is generated while the corresponding DMA channel n is held pending or while data is being written to the register, regardless of whether the DMA operation of DMA channel n is enabled or disabled, and regardless of the value set to the DTFRn register, is cleared.

4. If a DMA transfer start trigger is input while DMA is suspended (when the DCHC.ENn bit or DEN.ENnn bit = 0, or DCHC.STPn bit or DMSTP.STPnn bit = 1), the start trigger is held pending.

The pending start trigger is re-activated when the DMA operation is enabled (ENn/ENnn bit = 1, STPn/STPnn bit = 0) and DMA transfer is started.

5. Do not change the DTFRn register setting from the start of DMA transfer until the end of the specified number of DMA transfers. If this register setting is changed, the operation is not guaranteed.

6. While DMA transfer is pending because the bus mastership has been lost or because the transfer request has a low priority, only one start trigger is pended, even if two or more DMA start triggers have been generated and are waiting.

7. An interrupt request input from on-chip peripheral I/O during standby (IDLE or STOP mode) is held pending as a DMA transfer start trigger. The pending DMA start trigger is executed once the system returns to normal mode.

8. If the same start trigger is specified for multiple DMA channels, that DMA transfer start trigger will be enabled for all the specified channels at the same time. In this case, DMA transfer will be performed in order from the DMA channel with the highest priority.

Remark n = 0 to 6, m = 1 to 6, n < m (the priority of DMAn is higher than DMAm)

After reset: 0000H R/W Address: DTFR0 FFFFF080H,
DTFR0L FFFFF080H, DTFR0H FFFFF081H,
DTFR1 FFFFF090H,
DTFR1L FFFFF090H, DTFR1H FFFFF091H,
DTFR2 FFFFF0A0H,
DTFR2L FFFFF0A0H, DTFR2H FFFFF0A1H,
DTFR3 FFFFF0B0H,
DTFR3L FFFFF0B0H, DTFR3H FFFFF0B1H,
DTFR4 FFFFF0C0H,
DTFR4L FFFFF0C0H, DTFR4H FFFFF0C1H,
DTFR5 FFFFF0D0H,
DTFR5L FFFFF0D0H, DTFR5H FFFFF0D1H,
DTFR6 FFFFF0E0H,
DTFR6L FFFFF0E0H, DTFR6H FFFFF0E1H

	15	14	13	12	11	10	9	8
DTFRn (DTFRnH)	DFn	0	0	0	0	0	0	0
(n = 0 to 6)	7	6	5	4	3	2	1	0
(DTFRnL)	0	IFCn6	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

DFn[Note]	DMA transfer request status flag
0	No DMA transfer request/request cleared
1	DMA transfer request

- The DFn bit is used to check DMA transfer requests and to clear a request by setting the DFn bit = 0.
- This bit is cleared to 0 when DMA transfer starts in the single transfer mode, and it is cleared to 0 when the 1st DMA transfer starts in the single-step transfer mode.
- The DFn bit is set to 1 when a DMA start trigger is generated, regardless of the DCHCn.ENn or DEN.ENnn bit.

Note Do not set the DFn bit to 1 by software.
If an interrupt that is set as a DMA transfer start trigger is generated while DMA transfer is disabled (including forcible termination by software) and it is necessary to clear the DMA transfer request, write 0 to the DFn bit after stopping the operation that has caused the interrupt. If it is clear application-wise that an interrupt will not occur again until the next time DMA transfer resumes, the operation that caused the interrupt does not have to be stopped.

Caution For details about the IFCn6 to IFCn0 bits, see Table 20-4 DMA Transfer Start Triggers.

Table 20-4. DMA Transfer Start Triggers (1/3)

IFCn6	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt source
0	0	0	0	0	0	0	DMA transfer request from on-chip peripheral I/O disabled
0	0	0	0	0	0	1	INTLVIL
0	0	0	0	0	1	0	INTLVIH
0	0	0	0	0	1	1	INTP03
0	0	0	0	1	0	0	INTP04
0	0	0	0	1	0	1	INTP05
0	0	0	0	1	1	0	INTP06
0	0	0	0	1	1	1	INTP07
0	0	0	1	0	0	0	INTP08
0	0	0	1	0	0	1	INTP09
0	0	0	1	0	1	0	INTP10
0	0	0	1	0	1	1	INTP11
0	0	0	1	1	0	0	INTP12
0	0	0	1	1	0	1	INTP13
0	0	0	1	1	1	0	INTP14
0	0	0	1	1	1	1	INTTB0CC0
0	0	1	0	0	0	0	INTTB1CC0
0	0	1	0	0	0	1	INTTB0OV_BASE[Note]
0	0	1	0	0	1	0	INTTB0OV
0	0	1	0	0	1	1	INTTB1OV_BASE[Note]
0	0	1	0	1	0	0	INTTB1OV
0	0	1	0	1	0	1	INTCMP0L
0	0	1	0	1	1	0	INTCMP0F
0	0	1	0	1	1	1	INTCMP1L
0	0	1	1	0	0	0	INTCMP1F
0	0	1	1	0	0	1	INTTB0CC0_BASE[Note]
0	0	1	1	0	1	0	INTTB0CC1
0	0	1	1	0	1	1	INTTB0CC2
0	0	1	1	1	0	0	INTTB0CC3
0	0	1	1	1	0	1	INTTB1CC0_BASE[Note]
0	0	1	1	1	1	0	INTTB1CC1
0	0	1	1	1	1	1	INTTB1CC2
0	1	0	0	0	0	0	INTTB1CC3
0	1	0	0	0	0	1	INTTTIOV0
0	1	0	0	0	1	0	INTTTEQC00
0	1	0	0	0	1	1	INTTTEQC01
0	1	0	0	1	0	0	INTTTIOV1
0	1	0	0	1	0	1	INTTTEQC10
0	1	0	0	1	1	0	INTTTEQC11

Remark n = 0 to 6

Note INTTBaOV_BASE and INTTBaCC0_BASE are the INTTBaOV and INTTBaCC0 interrupt signals before they were culled by using the TMQa option (TMQOPa) in the 6-phase PWM output mode (a = 0, 1). For details, see **Figure 10-2 TMQn Option**.

Table 20-4. DMA Transfer Start Triggers (2/3)

IFCn6	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt source
0	1	0	0	1	1	1	INTTTIOV2
0	1	0	1	0	0	0	INTTTEQC20
0	1	0	1	0	0	1	INTTTEQC21
0	1	0	1	0	1	0	INTTTIOV3
0	1	0	1	0	1	1	INTTTEQC30
0	1	0	1	1	0	0	INTTTEQC31
0	1	0	1	1	0	1	INTTA0OV
0	1	0	1	1	1	0	INTTA0CC0
0	1	0	1	1	1	1	INTTA0CC1
0	1	1	0	0	0	0	INTTA1OV
0	1	1	0	0	0	1	INTTA1CC0
0	1	1	0	0	1	0	INTTA1CC1
0	1	1	0	0	1	1	INTTA2OV
0	1	1	0	1	0	0	INTTA2CC0
0	1	1	0	1	0	1	INTTA2CC1
0	1	1	0	1	1	0	INTDMA0
0	1	1	0	1	1	1	INTDMA1
0	1	1	1	0	0	0	INTDMA2
0	1	1	1	0	0	1	INTDMA3
0	1	1	1	0	1	0	INTDMA4
0	1	1	1	0	1	1	INTDMA5
0	1	1	1	1	0	0	INTUBTIR
0	1	1	1	1	0	1	INTUBTIT
0	1	1	1	1	1	0	INTUBTIF
0	1	1	1	1	1	1	INTUA0R
1	0	0	0	0	0	0	INTUA0T
1	0	0	0	0	0	1	INTCF0R
1	0	0	0	0	1	0	INTCF0T
1	0	0	0	0	1	1	INTUA1R
1	0	0	0	1	0	0	INTUA1T
1	0	0	0	1	0	1	INTCF1R
1	0	0	0	1	1	0	INTCF1T
1	0	0	0	1	1	1	INTUA2R
1	0	0	1	0	0	0	INTUA2T
1	0	0	1	0	0	1	INTCF2R
1	0	0	1	0	1	0	INTCF2T
1	0	0	1	0	1	1	INTIIC
1	0	0	1	1	0	0	INTAD0
1	0	0	1	1	0	1	INTAD1
1	0	0	1	1	1	0	INTAD2

Remark n = 0 to 6

Table 20-4. DMA Transfer Start Triggers (3/3)

IFCn6	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt source
1	0	0	1	1	1	1	INTTM0EQ0
1	0	1	0	0	0	0	INTTM1EQ0
1	0	1	0	0	0	1	INTTM2EQ0
1	0	1	0	0	1	0	INTTM3EQ0
1	0	1	0	0	1	1	INTDMA6
Other than above							Setting prohibited

Remark n = 0 to 6

20.4 Transfer Modes

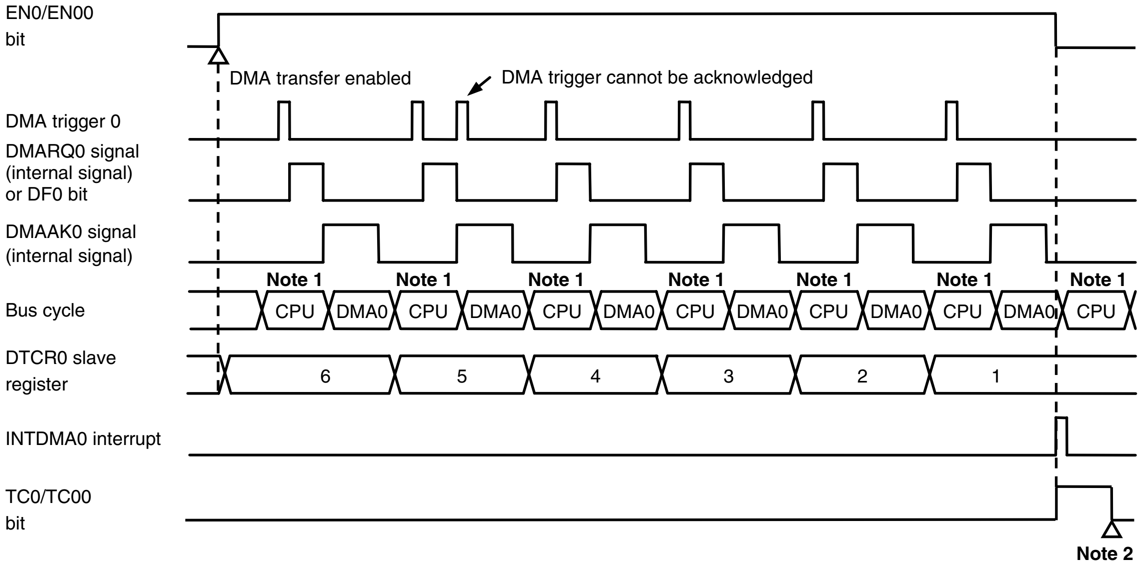
20.4.1 Single transfer mode

In single transfer mode, DMA transfer is performed once for each DMA transfer request. If there is a subsequent DMA transfer request, DMA transfer is performed again. This operation continues until a terminal count occurs (at the end of the specified number of DMA transfers). The DMAC releases the bus after each DMA transfer cycle.

If, after the DMAC has released the bus, another higher priority DMA transfer request is issued, the higher priority DMA transfer request always takes precedence for the next DMA transfer. In addition, if DMA triggers for two or more channels occur at the same time, the higher priority DMA transfer request is given priority. However, if another DMA transfer request with a lower priority occurs within one clock of the completion of single transfer, even if the previous higher priority DMA transfer request signal remains active, this higher priority DMA transfer request does not take precedence. The lower priority, newly requested DMA transfer will be executed after the bus has been released to the CPU.

Examples of single transfer are shown below.

Figure 20-2. Single Transfer Example 1



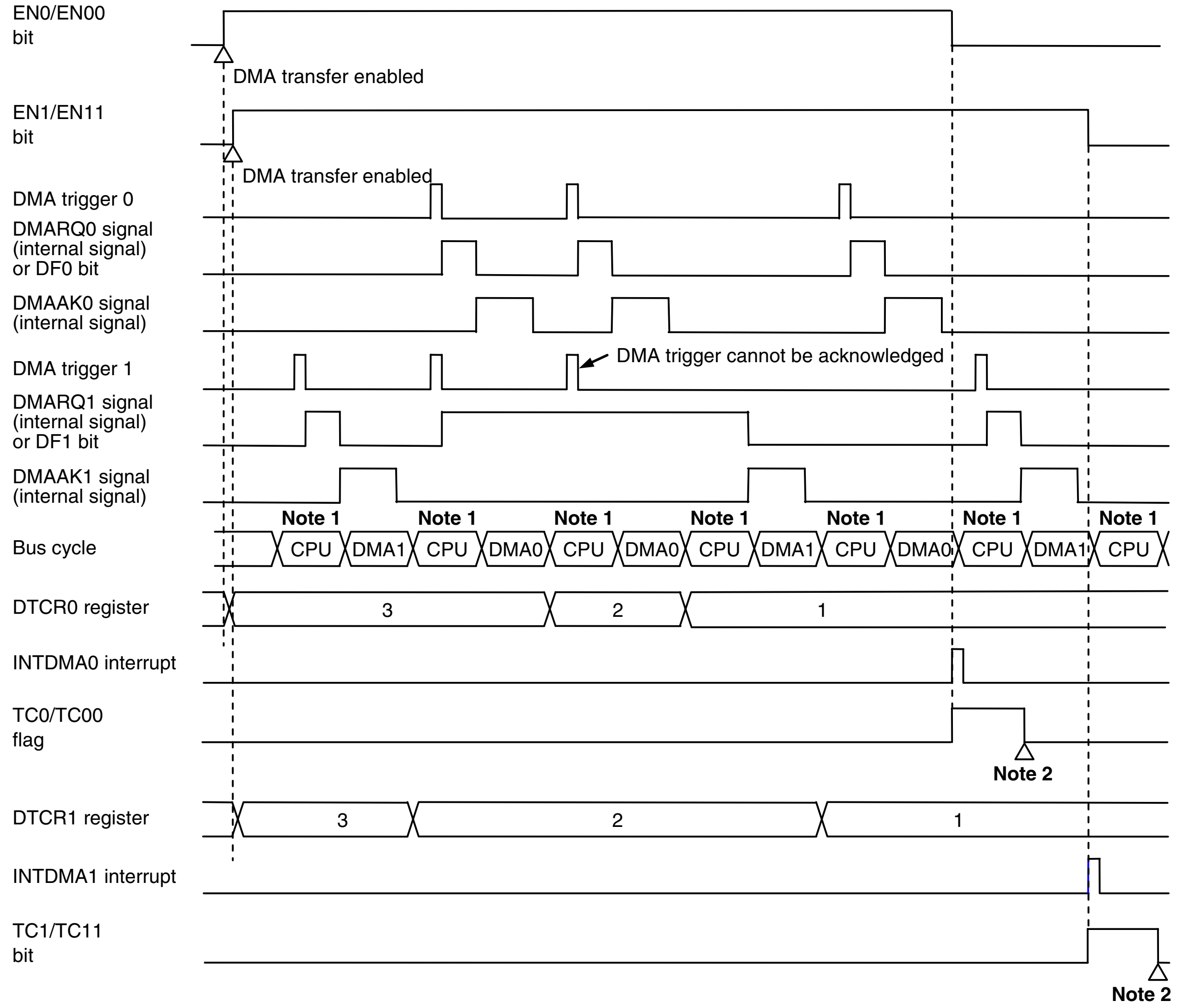
Notes 1. The bus is always released.

2. The DCHC0 register is read or “0” is written to the DMAS.TC00 bit.

Caution During the period from DMA trigger generation to the falling edge of the DMAAK0 signal (internal signal), DMA triggers for the same channel cannot be acknowledged.

Remark When the active level of the DMAAK0 signal (internal signal) is the high level.

Figure 20-3. Single Transfer Example 2



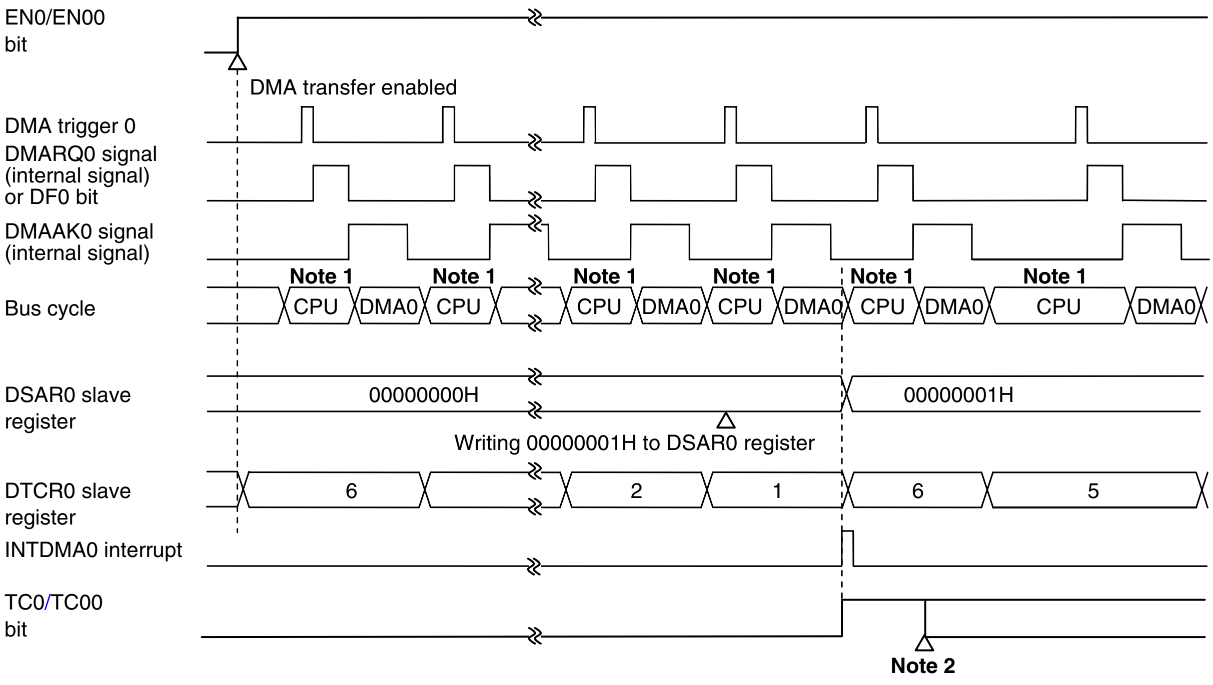
Notes 1. The bus is always released.

2. The DCHCn register is read or writing “0” to the DMAS.TCnn bit

Caution **During the period from DMA trigger generation to the falling edge of the DMAAK0 and DMAAK1 signals (internal signals), DMA triggers for the same channel cannot be acknowledged.**

Remark When the active level of the DMAAK0 and DMAAK1 signals (internal signals) is the high level.

Figure 20-4. Single Transfer Example 3 (with Next Address Setting Function Enabled)



Notes 1. The bus is always released.

2. The DCHC0 register is read or writing “0” to the DMAS.TC00 bit

Caution **During the period from DMA trigger generation to the falling edge of the DMAAK0 signal (internal signal), DMA triggers for the same channel cannot be acknowledged.**

Remark When the active level of the DMAAK0 signal (internal signal) is the high level.

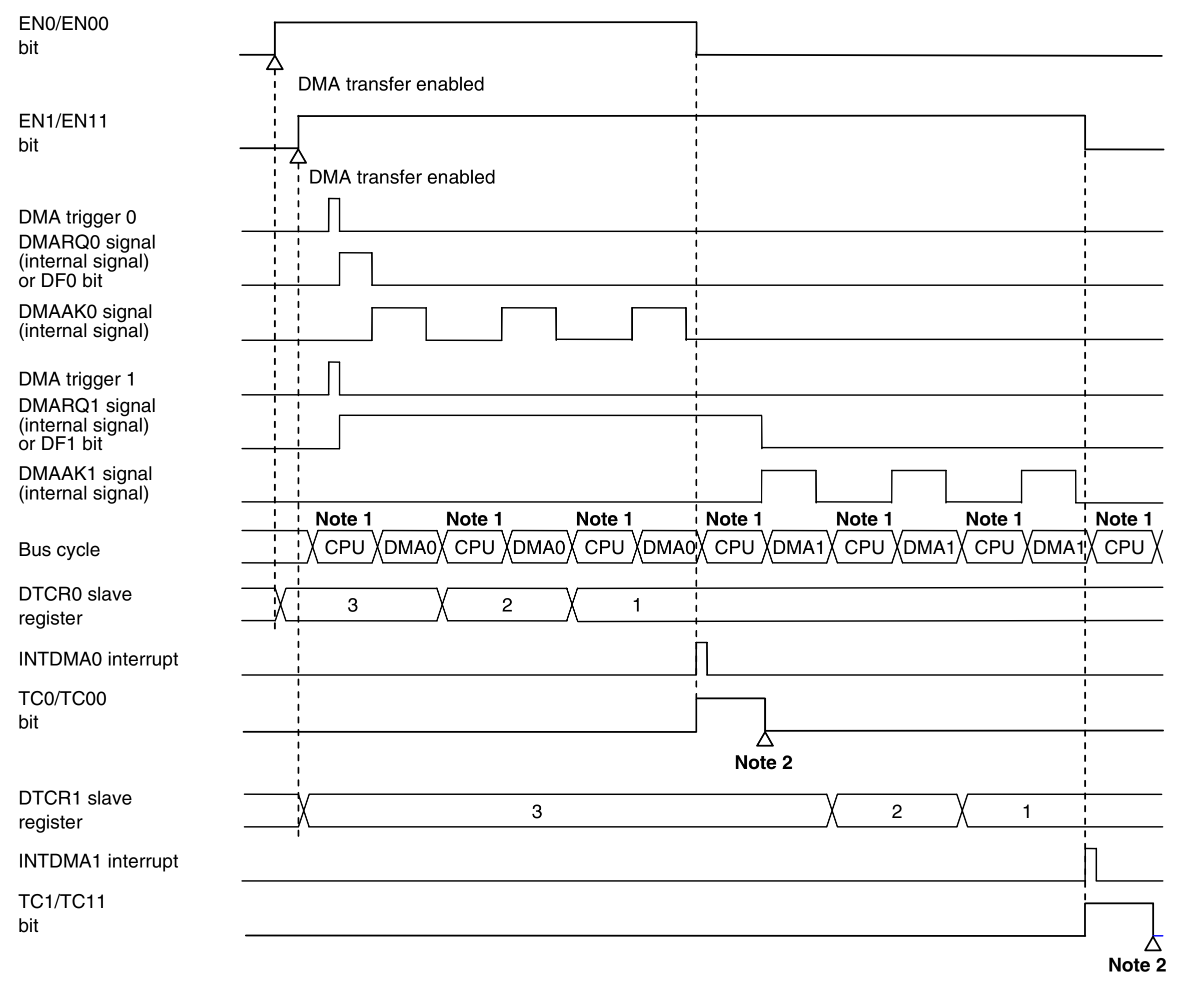
20.4.2 Single-step transfer mode

In the single-step transfer mode, DMA transfer is executed a specified number of times in response to one DMA transfer request. When a DMA transfer request is acknowledged, the bus is released after each DMA transfer cycle and the DMA operation continues until a terminal count occurs (i.e., when transfer has been completed the specified number of times).

If, after the DMAC has released the bus, another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence for the next DMA transfer. In addition, if DMA triggers for two or more channels occur at the same time, the higher priority DMA transfer request is given priority.

Figures 20-5 and 20-6 show examples of single-step transfer.

Figure 20-5. Single-Step Transfer Example 1



Notes 1. The bus is always released.

2. The DCHCn register is read or “0” is written to the DMAS.TCnn bit

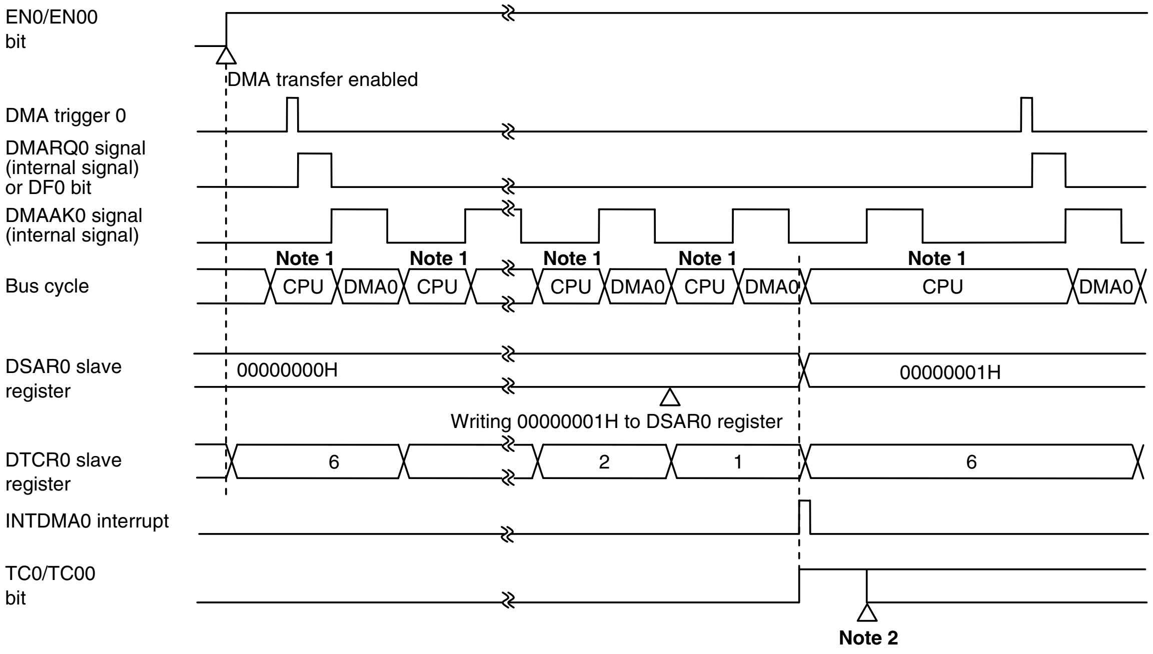
Caution During the period from DMA trigger generation to the falling edge of the DMAAK0 and DMAAK1 signals (internal signals), DMA triggers for the same channel cannot be acknowledged.

Remarks 1. DMA is started when the DCHC.STGn bit is set to 1 while DMA transfer is enabled (DCHC.TCn bit or DMAS.TCnn bit = 0 and DCHC.ENn bit or DEN.ENnn bit = 1).

2. When the active level of the DMAAK0 and DMAAK1 signals (internal signals) is the high level.

3. n = 0, 1

Figure 20-6. Single-Step Transfer Example 2 (with Next Address Setting Function Enabled)



Notes 1. The bus is always released.

2. The DCHC0 register is read or writing “0” to the DMAS.TC00 bit

Caution **During the period from DMA trigger generation to the falling edge of the DMAAK0 pin, DMA triggers for the same channel cannot be acknowledged.**

Remarks 1. DMA is started when the DCHC.STG0 bit is set to 1 while DMA transfer is enabled (DCHC.TC0 bit or DMAS.TC00 bit = 0 and DCHC.ENn bit or DEN.EN00 bit = 1).

2. When the active level of the DMAAK0 signal (internal signal) is the high level.

20.5 Transfer Types

Two-cycle transfer is supported as the DMA transfer type. In two-cycle transfer, data transfer is performed in two cycles, a read cycle (transfer source to DMAC) and a write cycle (DMAC to transfer destination).

In the first cycle, the transfer source address is output and data is read from the source to the DMAC. In the second cycle, the destination address is output and the data is written from the DMAC.

20.6 Transfer Sources and Destinations

The following list shows the compatibility of various transfer sources and destinations (√: Transfer enabled, ×: Transfer disabled).

Table 20-5. Relationship Between Transfer Sources and Destinations

		Destination		
		Internal ROM	On-Chip Peripheral I/O	Internal ROM
Source	Internal RAM	×	√	×
	On-chip peripheral I/O	√	×	×
	Internal ROM	×	×	×

Caution Transfer is not guaranteed for transfer destination and transfer source combinations marked with "×" in Table 20-5.

20.7 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

Table 20-6. DMA Priorities

DMA Channel	Priority
Channel 0	Highest
Channel 1	↕
Channel 2	
Channel 3	
Channel 4	
Channel 5	
Channel 6	Lowest

In the single-step transfer mode, if a higher priority DMA transfer request is issued while the bus is released, the higher priority DMA transfer request is acknowledged and executed.

If the same start triggers are allocated to two or more DMA channels, the higher priority DMA channel is acknowledged before the lower priority DMA channels.

20.8 Next Address Setting Function

The DMA transfer source address specification register n (DSARn, DSARnH, DSARnL), DMA transfer destination address specification register n (DDARn, DDARnH, DDARnL) and DMA transfer count specification register n (DTCRn) are 2-stage FIFO buffer registers that consist of a master register and a slave register.

When a terminal count occurs (at the end of the specified number of DMA transfers), the master register value is transferred to the slave register.

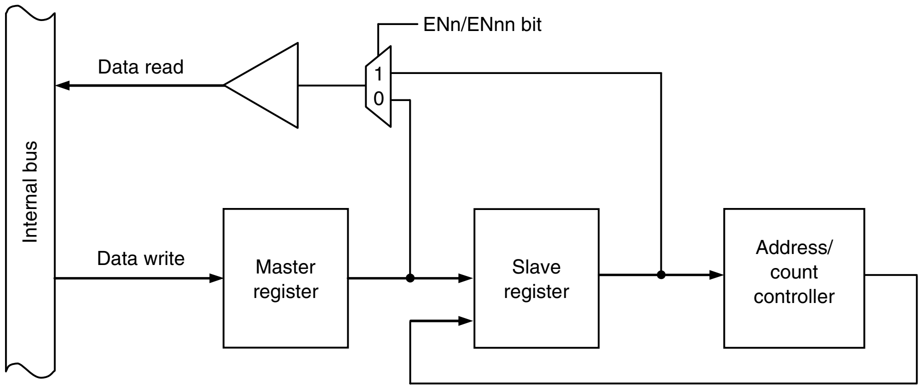
Therefore, if a new DMA transfer setting is performed for these registers during DMA transfer, values are automatically updated to the new values after the transfer is complete.

Remark n = 0 to 6

20.9 Buffer Register Configuration

The figure below shows the configuration of the buffer register.

Figure 20-7. Buffer Register Configuration



The set values for DMA transfer source address specification register n (DSARn, DSARnH, DSARnL), DMA transfer destination address specification register n (DDARn, DDARnH, DDARnL), and DMA transfer count specification register n (DTCRn) are applied to the master register.

These values are then applied to the slave register when the DCHC.ENn bit or the DEN.ENnn bit changes from “0” to “1”. When the next address function is used, the contents of the master register are transferred to the slave register when the DCHC.TCn bit or the DMAS.TCnn bit = 1. The actual DMA transfer is performed based on the slave register setting.

Cautions 1. To set a new DMA transfer when the next address function is used, write the master register when the ENn/ENnn bit = 1 and before generation of a terminal count.

2. When the next address function is used and when DMA transfer has been completed the specified number of times, the TCn/TCnn bit is set to 1, but the ENn/ENnn bit is not cleared to 0 and remains “1”.

3. To set the DCHC.STGn bit to the specified number of cycles of DMA transfer when the next address function is used, clear the TCn/TCnn bit to 0 after DMA transfer is completed (TCn/TCnn bit = 1) and then set the STGn bit to 1.

Remarks 1. “1” is written to the ENn/ENnn bit while the bit is “1”, data is not transferred to the slave register.

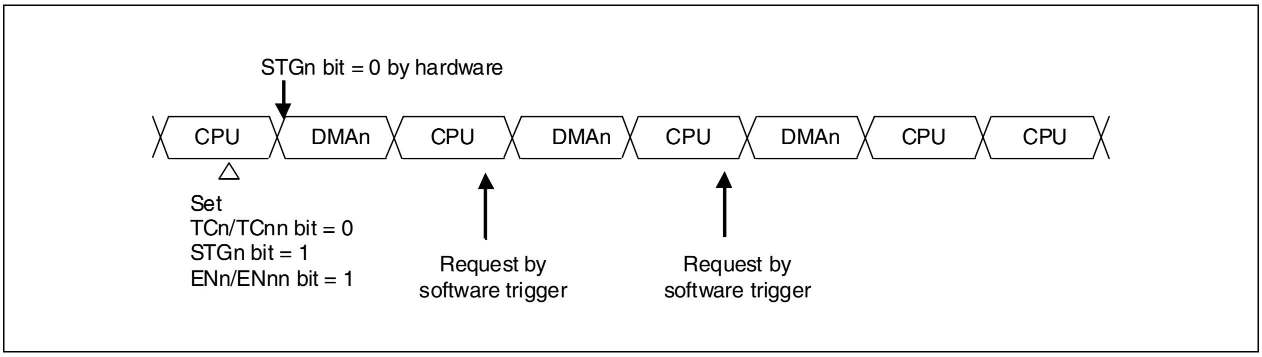
2. n = 0 to 6

20.10 DMA Transfer Start Triggers

There are two types of DMA transfer start triggers, as shown below.

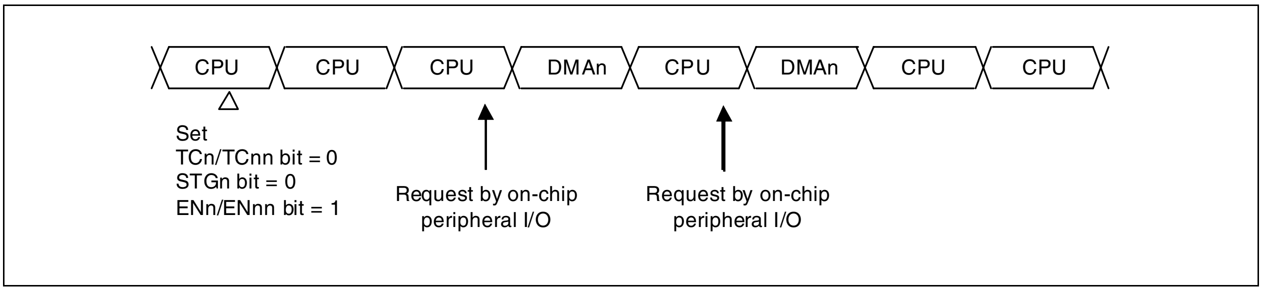
(1) Transfer start triggered by software

When the DCHCn.TCn bit or the DMAS.TCnn bit is set to 0, the DCHCn.STGn bit is set to 1, and the DCHCn.ENn bit or DEN.ENnn bit is set to 1, DMA transfer is started by software request.



(2) Transfer start triggered by request from on-chip peripheral I/O

If an interrupt request is generated from an on-chip peripheral I/O set in the DTFRn register when the DCHCn.TCn bit or the DMAS.TCnn bit is 0, the DCHCn.STGn bit is 0, and the DCHCn.ENn bit or the DEN.ENnn bit is 1, DMA transfer is started.



Remarks 1. When using the next address function, DMA transfer is started even when the TCn/TCnn bit = 1.

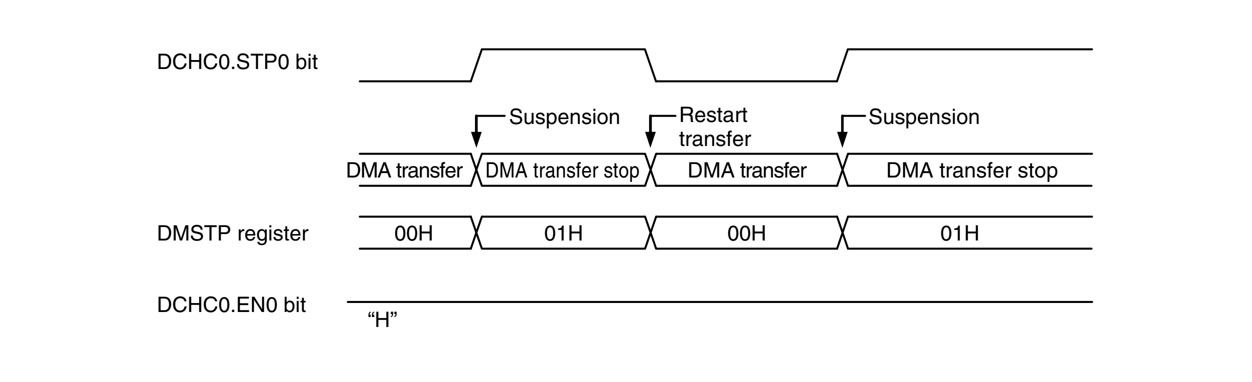
2. n = 0 to 6

20.11 Suspension

DMA transfer can be suspended by setting the DCHCn.STPn bit or the DMSTP.STPnn bit to 1 during DMA transfer. If the next DMA transfer request has already been acknowledged at this time, the next DMA transfer bus cycle is completed first, and then DMA transfer is suspended. Channels for which DMA transfer has been suspended are excluded from priority judgment.

The DMA transfer request is retained in the DMAC, and when the STPn/STPnn bit is cleared to 0, DMA transfer will restart from the next transfer after the transfer that was suspended.

Figure 20-8. Example of DMA Transfer Suspension (DMA0)



During single transfer, only one suspended DMA transfer start trigger can be held pending (even if two or more start triggers have been generated), and transfer is restarted after the STPn/STPnn bit is set to 0.

During single-step transfer, the suspended DMA transfer start trigger is ignored without being held pending. At this time, the DFn bit = 1 and is cleared to 0 when DMA transfer is restarted.

Remark n = 0 to 6

20.12 End of DMA Transfer

A DMA transfer end interrupt (INTDMAn) is generated when a terminal count occurs (at the end of the specified number of DMA transfers). If this time, the DCHCn.TCn bit or the DMAS.TCnn bit is set to 1.

Remark n = 0 to 6

20.13 Forcible Termination

The DMA transfer of DMA channel n can be forcibly terminated by clearing the DCHCn.ENn bit or the DEN.ENnn bit to 0. If DMA transfer is in progress at this time, the forcible termination is executed at the end of the current bus cycle. If the ENn/ENnn bit is still "0" after writing 0 to the ENn/ENnn bit, forcible termination is complete.

In the case of forcible termination, a DMA transfer end interrupt (INTDMAn) is not generated.

Remark n = 0 to 6

20.14 Cautions

(1) Memory boundary

Transfer is not guaranteed if the transfer source or the transfer destination address exceeds the DMA area (internal RAM or on-chip peripheral I/O) during DMA transfer.

(2) Transfer of misaligned data

32-bit or 16-bit misaligned data cannot be transferred by DMA.

(3) Bus arbitration for CPU

Because the DMAC takes precedence over the CPU in acquiring bus mastership, if the CPU requests access during DMA transfer, the CPU will not be able to execute the access until the DMA transfer is completed and the bus is released to the CPU.

However, the CPU can access the internal ROM and internal RAM, as long as they are not being accessed by the DMAC.

The CPU can also access the internal ROM while the DMAC is executing DMA transfer between an on-chip peripheral I/O and the internal RAM.

(4) Program execution in internal RAM and DMA transfer

The CPU may deadlock under the following conditions. In this case, the only action that can be executed is a reset.

Conditions

A DMA transfer to transfer data to/from the internal RAM is executed while any of the following instructions is being executed.

- A bit manipulation instruction located in the internal RAM (SET1, CLR1, NOT1)
- A data access instruction that accesses a misaligned address located in the internal RAM

Measures this problem can be avoided by taking either of the following measures.

Measures

- Do not execute a bit manipulation instruction (SET1, CLR1, or NOT1) located in the internal RAM or a data access instruction that accesses a misaligned address when DMA transfer is being executed to transfer data to/from the internal RAM.
- Do not execute DMA transfer that transfers data to/from the internal RAM when a bit manipulation instruction (SET1, CLR1, or NOT1) located in the internal RAM or a data access instruction that accesses a misaligned address is being executed.

(5) Delay in start of DMA transfer

The start of DMA transfer may be delayed by an internal RAM access or on-chip peripheral I/O access by the CPU.

(6) Special register settings while using DMA

See **3.4.8 (1) Setting data to special registers**.

(7) Registers that must not be set under certain conditions

The following registers must not be written to when a specific operation is performed. If these registers are written to, the operation cannot be guaranteed.

Status	Register That Must Not Be Set
Stop (ENn/ENnn bit = 0)	None
During operation (ENn/ENnn bit = 1)	DADCn[Note 1], DTFRn
During suspension[Note 2] (STPn/STPnn bit = 1)	DADCn[Note 1], DTFRn

Notes 1. The same value may be written to the register.

2. Setting the register is prohibited when the operation that was suspended is resumed. The register can be set when the operation is stopped (ENn/ENnn bit = 0) after the register is written.

Remark n = 0 to 6

CHAPTER 21 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION

The V850E/IG4-H and V850E/IH4-H are provided with an interrupt controller dedicated to interrupt servicing (INTC) and can handle a total of 106 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.

The V850E/IG4-H and V850E/IH4-H can handle interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

21.1 Features

○ Interrupts

- Non-maskable interrupts: 1 source (external: none, internal: 1 source)
- Maskable interrupts (the number of maskable interrupt sources differs depending on the product)
  105 sources (external: 22 sources, internal: 83 sources)
- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals

○ Exceptions

- Software exceptions: 32 sources
- Exception traps: 2 sources (illegal opcode exception, debug trap)

The interrupt sources are listed in Table 21-1.

Table 21-1. Interrupt Source List (1/4)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Return PC
		Name	Control Register	Generating Source	Generating Unit				
Reset	Interrupt	RESET	–	$\overline{\text{RESET}}$ pin input Reset input from internal source	Pin/WDT/ LVI/POC	–	0000H	00000000H	Undefined
Non-maskable	Interrupt	INTWDT	–	WDT overflow	WDT	–	0010H	00000010H	Next PC
Software exception	Exception	TRAP0n[Note]	–	TRAP instruction	–	–	004nH	00000040H	Next PC
	Exception	TRAP1n[Note]	–	TRAP instruction	–	–	005nH	00000050H	Next PC
Exception trap	Exception	ILGOP/ DBG0	–	Illegal instruction code/ DBTRAP instruction	–	–	0060H	00000060H	Next PC
Maskable	Interrupt	INTLVIL	LVILIC	LVI low level voltage detection	LVI	0	0080H	00000080H	Next PC
	Interrupt	INTLVIH	LVIHIC	LVI high level voltage detection	LVI	1	0090H	00000090H	Next PC
	Interrupt	INTP00	PIC00	INTP00 pin valid edge input	Pin	2	00A0H	000000A0H	Next PC
	Interrupt	INTP01	PIC01	INTP01 pin valid edge input	Pin	3	00B0H	000000B0H	Next PC
	Interrupt	INTP02	PIC02	INTP02 pin valid edge input	Pin	4	00C0H	000000C0H	Next PC
	Interrupt	INTP03	PIC03	INTP03 pin valid edge input	Pin	5	00D0H	000000D0H	Next PC
	Interrupt	INTP04	PIC04	INTP04 pin valid edge input	Pin	6	00E0H	000000E0H	Next PC
	Interrupt	INTP05	PIC05	INTP05 pin valid edge input	Pin	7	00F0H	000000F0H	Next PC
	Interrupt	INTP06	PIC06	INTP06 pin valid edge input	Pin	8	0100H	00000100H	Next PC
	Interrupt	INTP07	PIC07	INTP07 pin valid edge input	Pin	9	0110H	00000110H	Next PC
	Interrupt	INTP08	PIC08	INTP08 pin valid edge input	Pin	10	0120H	00000120H	Next PC
	Interrupt	INTP09	PIC09	INTP09 pin valid edge input	Pin	11	0130H	00000130H	Next PC
	Interrupt	INTP10	PIC10	INTP10 pin valid edge input	Pin	12	0140H	00000140H	Next PC
	Interrupt	INTP11	PIC11	INTP11 pin valid edge input	Pin	13	0150H	00000150H	Next PC
	Interrupt	INTP12	PIC12	INTP12 pin valid edge input	Pin	14	0160H	00000160H	Next PC
	Interrupt	INTP13	PIC13	INTP13 pin valid edge input	Pin	15	0170H	00000170H	Next PC
	Interrupt	INTP14	PIC14	INTP14 pin valid edge input	Pin	16	0180H	00000180H	Next PC
	Interrupt	INTP15	PIC15	INTP15 pin valid edge input	Pin	17	0190H	00000190H	Next PC
	Interrupt	INTP16	PIC16	INTP16 pin valid edge input	Pin	18	01A0H	000001A0H	Next PC
	Interrupt	INTP17	PIC17	INTP17 pin valid edge input	Pin	19	01B0H	000001B0H	Next PC
	Interrupt	INTP18	PIC18	INTP18 pin valid edge input	Pin	20	01C0H	000001C0H	Next PC
	Interrupt	INTP19	PIC19	INTP19 pin valid edge input	Pin	21	01D0H	000001D0H	Next PC
	Interrupt	INTCMP0L	CMPIC0L	ADC0 overvoltage detection L (comparator output)	ADC0 (comparator)	22	01E0H	000001E0H	Next PC
	Interrupt	INTCMP0F	CMPIC0F	ADC0 overvoltage detection F (comparator output)	ADC0 (comparator)	23	01F0H	000001E0H	Next PC
	Interrupt	INTCMP1L	CMPIC1L	ADC1 overvoltage detection L (comparator output)	ADC1 (comparator)	24	0200H	00000200H	Next PC
	Interrupt	INTCMP1F	CMPIC1F	ADC1 overvoltage detection F (comparator output)	ADC1 (comparator)	25	0210H	00000210H	Next PC

Note n = 0 to FH

Table 21-1. Interrupt Source List (2/4)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Return PC
		Name	Control Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTTB0OV	TB0OVIC	TAB0 overflow[Note 1]	TMQOP0	26	0220H	00000220H	Next PC
	Interrupt	INTTB0CC0	TB0CCIC0	TAB0CCR0 capture input/ compare match[Note 2]	TMQOP0	27	0230H	00000230H	Next PC
	Interrupt	INTTB0CC1	TB0CCIC1	TAB0CCR1 capture input/ compare match	TAB0	28	0240H	00000240H	Next PC
	Interrupt	INTTB0CC2	TB0CCIC2	TAB0CCR2 capture input/ compare match	TAB0	29	0250H	00000250H	Next PC
	Interrupt	INTTB0CC3	TB0CCIC3	TAB0CCR3 capture input/ compare match	TAB0	30	0260H	00000260H	Next PC
	Interrupt	INTTB1OV	TB1OVIC	TAB1 overflow[Note 1]	TMQOP1	31	0270H	00000270H	Next PC
	Interrupt	INTTB1CC0	TB1CCIC0	TAB1CCR0 capture input[Note 3]/ compare match[Note 2]	TMQOP1	32	0280H	00000280H	Next PC
	Interrupt	INTTB1CC1	TB1CCIC1	TAB1CCR1 capture input[Note 3]/ compare match	TAB1	33	0290H	00000290H	Next PC
	Interrupt	INTTB1CC2	TB1CCIC2	TAB1CCR2 capture input[Note 3]/ compare match	TAB1	34	02A0H	000002A0H	Next PC
	Interrupt	INTTB1CC3	TB1CCIC3	TAB1CCR3 capture input[Note 3]/ compare match	TAB1	35	02B0H	000002B0H	Next PC
	Interrupt	INTTTIOV0	TT0OVIC	TMT0 overflow	TMT0	36	02C0H	000002C0H	Next PC
	Interrupt	INTTTEQC00	TT0CCIC0	TT0CCR0 capture input/ compare match	TMT0	37	02D0H	000002D0H	Next PC
	Interrupt	INTTTEQC01	TT0CCIC1	TT0CCR1 capture input/ compare match	TMT0	38	02E0H	000002E0H	Next PC
	Interrupt	INTTIEC0	TT0IECIC	Encoder input interrupt 0	TMT0	39	02F0H	000002F0H	Next PC
	Interrupt	INTTTIOV1	TT1OVIC	TMT1 overflow	TMT1	40	0300H	00000300H	Next PC
	Interrupt	INTTTEQC10	TT1CCIC0	TT1CCR0 capture input/ compare match	TMT1	41	0310H	00000310H	Next PC
	Interrupt	INTTTEQC11	TT1CCIC1	TT1CCR1 capture input/ compare match	TMT1	42	0320H	00000320H	Next PC
	Interrupt	INTTIEC1	TT1IECIC	Encoder input interrupt 1	TMT1	43	0330H	00000330H	Next PC
	Interrupt	INTTA0OV2	TT2OVIC	TMT2 overflow	TMT2	44	0340H	00000340H	Next PC

Notes 1. When TABm is used in the 6-phase PWM output mode, it functions as INTTBmOV (trough interrupt) from the TMQm option (TMQOPm) (m = 0, 1).

2. When TABm is used in the 6-phase PWM output mode, it functions as INTTBmCC0 (peak interrupt) from the TMQm option (TMQOPm) (m = 0, 1).

3. V850E/IH4-H only
Only a compare match is available for the V850E/IG4-H

Table 21-1. Interrupt Source List (3/4)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Return PC
		Name	Control Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTTTEQC20	TT2CCIC0	TT2CCR0 capture input/ compare match	TMT2	45	0350H	00000350H	Next PC
	Interrupt	INTTTEQC21	TT2CCIC1	TT2CCR1 capture input/ compare match	TMT2	46	0360H	00000360H	Next PC
	Interrupt	INTTTIOV3	TT3OVIC	TMT3 overflow	TMT3	47	0370H	00000370H	Next PC
	Interrupt	INTTTEQC30	TT3CCIC0	TT3CCR0 capture input/ compare match	TMT3	48	0380H	00000380H	Next PC
	Interrupt	INTTTEQC31	TT3CCIC1	TT3CCR1 capture input/ compare match	TMT3	49	0390H	00000390H	Next PC
	Interrupt	INTTA0OV	TA0OVIC	TAA0 overflow	TAA0	50	03A0H	000003A0H	Next PC
	Interrupt	INTTA0CC0	TA0CCIC0	TA0CCR0 compare match	TAA0	51	03B0H	000003B0H	Next PC
	Interrupt	INTTA0CC1	TA0CCIC1	TA0CCR1 compare match	TAA0	52	03C0H	000003C0H	Next PC
	Interrupt	INTTA1OV	TA1OVIC	TAA1 overflow	TAA1	53	03D0H	000003D0H	Next PC
	Interrupt	INTTA1CC0	TA1CCIC0	TA1CCR0 compare match	TAA1	54	03E0H	000003E0H	Next PC
	Interrupt	INTTA1CC1	TA1CCIC1	TA1CCR1 compare match	TAA1	55	03F0H	000003F0H	Next PC
	Interrupt	INTTA2OV	TA2OVIC	TAA2 overflow	TAA2	56	0400H	00000400H	Next PC
	Interrupt	INTTA2CC0	TA2CCIC0	TA2CCR0 capture input/ compare match	TAA2	57	0410H	00000410H	Next PC
	Interrupt	INTTA2CC1	TA2CCIC1	TA2CCR1 capture input/ compare match	TAA2	58	0420H	00000420H	Next PC
	Interrupt	INTDMA0	DMAIC0	DMA channel 0 transfer end	DMA0	59	0430H	00000430H	Next PC
	Interrupt	INTDMA1	DMAIC1	DMA channel 1 transfer end	DMA1	60	0440H	00000440H	Next PC
	Interrupt	INTDMA2	DMAIC2	DMA channel 2 transfer end	DMA2	61	0450H	00000450H	Next PC
	Interrupt	INTDMA3	DMAIC3	DMA channel 3 transfer end	DMA3	62	0460H	00000460H	Next PC
	Interrupt	INTDMA4	DMAIC4	DMA channel 4 transfer end	DMA4	63	0470H	00000470H	Next PC
	Interrupt	INTDMA5	DMAIC5	DMA channel 5 transfer end	DMA5	64	0480H	00000480H	Next PC
	Interrupt	INTUBTIRE	UREIC	UARTB reception error	UARTB	65	0490H	00000490H	Next PC
	Interrupt	INTUBTIF	URIC	UARTB reception end	UARTB	66	04A0H	000004A0H	Next PC
	Interrupt	INTUBTIT	UTIC	UARTB transmission enable	UARTB	67	04B0H	000004B0H	Next PC
	Interrupt	INTUBTIF	UIFIC	UARTB FIFO transmission end	UARTB	68	04C0H	000004C0H	Next PC
	Interrupt	INTUBTITO	UTOIC	UARTB reception timeout	UARTB	69	04D0H	000004D0H	Next PC
	Interrupt	INTUA0RE	UA0REIC	UARTA0 reception error	UARTA0	70	04E0H	000004E0H	Next PC
	Interrupt	INTUA0R	UA0RIC	UARTA0 reception end	UARTA0	71	04F0H	000004F0H	Next PC
	Interrupt	INTUA0T	UA0TIC	UARTA0 transmission enable	UARTA0	72	0500H	00000500H	Next PC
	Interrupt	INTCF0RE	CFOREIC	CSIF0 reception error	CSIF0	73	0510H	00000510H	Next PC
	Interrupt	INTCF0R	CF0RIC	CSIF0 reception end	CSIF0	74	0520H	00000520H	Next PC
	Interrupt	INTCF0T	CF0TIC	CSIF0 transmission enable	CSIF0	75	0530H	00000530H	Next PC
	Interrupt	INTUA1RE	UA1REIC	UARTA1 reception error	UARTA1	76	0540H	00000540H	Next PC
	Interrupt	INTUA1R	UA1RIC	UARTA1 reception end	UARTA1	77	0550H	00000550H	Next PC
	Interrupt	INTUA1T	UA1TIC	UARTA1 transmission enable	UARTA1	78	0560H	00000560H	Next PC

Table 21-1. Interrupt Source List (4/4)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Return PC
		Name	Control Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTCF1RE	CF1REIC	CSIF1 reception error	CSIF1	79	0570H	00000570H	Next PC
	Interrupt	INTCF1R	CF1RIC	CSIF1 reception end	CSIF1	80	0580H	00000580H	Next PC
	Interrupt	INTCF1T	CF1TIC	CSIF1 transmission enable	CSIF1	81	0590H	00000590H	Next PC
	Interrupt	INTUA2RE	UA2REIC	UARTA2 reception error	UARTA2	82	05A0H	000005A0H	Next PC
	Interrupt	INTUA2R	UA2RIC	UARTA2 reception end	UARTA2	83	05B0H	000005B0H	Next PC
	Interrupt	INTUA2T	UA2TIC	UARTA2 transmission enable	UARTA2	84	05C0H	000005C0H	Next PC
	Interrupt	INTCF2RE	CF2REIC	CSIF2 reception error	CSIF2	85	05D0H	000005D0H	Next PC
	Interrupt	INTCF2R	CF2RIC	CSIF2 reception end	CSIF2	86	05E0H	000005E0H	Next PC
	Interrupt	INTCF2T	CF2TIC	CSIF2 transmission enable	CSIF2	87	05F0H	000005F0H	Next PC
	Interrupt	INTIIC	IICIC	IIC serial transfer end	IIC	88	0600H	00000600H	Next PC
	Interrupt	INTAD0	AD0IC	ADC0 conversion end	ADC0	89	0610H	00000610H	Next PC
	Interrupt	INTAD1	AD1IC	ADC1 conversion end	ADC1	90	0620H	00000620H	Next PC
	Interrupt	INTAD2	AD2IC	ADC2 conversion end	ADC2	91	0630H	00000630H	Next PC
	Interrupt	INTTM0EQ0	TM0EQIC0	TM0CMP0 compare match	TMM0	92	0640H	00000640H	Next PC
	Interrupt	INTTM1EQ0	TM1EQIC0	TM1CMP0 compare match	TMM1	93	0650H	00000650H	Next PC
	Interrupt	INTTM2EQ0	TM2EQIC0	TM2CMP0 compare match	TMM2	94	0660H	00000660H	Next PC
	Interrupt	INTTM3EQ0	TM3EQIC0	TM3CMP0 compare match	TMM3	95	0670H	00000670H	Next PC
	Interrupt	INTADT0	ADT0IC	ADTRG0 pin valid edge input	Pin	96	0680H	00000680H	Next PC
	Interrupt	INTADT1	ADT1IC	ADTRG1 pin valid edge input	Pin	97	0690H	00000690H	Next PC
	Interrupt	INTUSBF0	UFIC0	USBF interrupt	USBF	98	06A0H	000006A0H	Next PC
	Interrupt	INTUSBF1	UFIC1	USBF resume interrupt	USBF	99	06B0H	000006B0H	Next PC
	Interrupt	INTDMA6	DMAIC6	DMA channel 6 transfer end	DMA	100	06C0H	000006C0H	Next PC
	Interrupt	INTTB0OV_BASE	TB0OVBIC	TAB0 overflow[Note 1]	TAB0	101	06D0H	000006D0H	Next PC
	Interrupt	INTTB0CC0_BASE	TB0CCBIC0	TAB0CCR0 capture input/ compare match[Note 2]	TAB0	102	06E0H	000006E0H	Next PC
	Interrupt	INTTB1OV_BASE	TB1OVBIC	TAB1 overflow[Note 1]	TAB1	103	06F0H	000006F0H	Next PC
	Interrupt	INTTB1CC0_BASE	TB1CCBIC0	TAB1CCR0 capture input/ compare match[Note 2]	TAB1	104	0700H	00000700H	Next PC

Notes 1. INTTBmOV_BASE is the INTTBmOV interrupt signal before it was culled by using the TMQm option (TMQOPm) in the 6-phase PWM output mode (m = 0, 1). For details, see Figure 10-2 TMQn Option.

2. INTTBmCC0_BASE is the INTTBmCC0 interrupt signal before it was culled by using the TMQm option (TMQOPm) in the 6-phase PWM output mode (m = 0, 1). For details, see Figure 10-2 TMQn Option.

Remarks 1. Default priority: The priority order that is applied when multiple maskable interrupt requests having the same priority level occur simultaneously. The highest priority is 0.

Return PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC of the CPU when interrupt servicing is started. Note, however, that the return PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the next PC. (If an interrupt is acknowledged during interrupt execution, execution stops, and then resumes after the interrupt servicing has finished. In this case, the address of the aborted instruction is the return PC.)

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

Next PC: The PC value that starts the processing following interrupt/exception servicing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (return PC – 4).

21.2 Non-Maskable Interrupts

A non-maskable interrupt request signal is acknowledged unconditionally, even when interrupts are disabled (DI) by the CPU. A non-maskable interrupt not subject to priority control and takes precedence over all the other interrupt request signals.

The V850E/IG4-H and V850E/IH4-H have one non-maskable interrupt signal which is the non-maskable interrupt request signal generated by the overflow of the watchdog timer (INTWDT).

INTWDT can be generated when the WDTM.WDM1 and WDTM.WDM0 bits are set to “01”.

21.2.1 Operation

If a non-maskable interrupt request signal (INTWDT) is generated, the CPU performs the following processing and transfers control to the handler routine.

(1) Saves the current PC to FEPC.
(2) Saves the current PSW to FEPSW.
(3) Writes the exception code (0010H) to the higher halfword (FECC) of ECR.
(4) Sets the PSW.NP and PSW.ID bits (1) and clears the PSW.EP bit (0).
(5) Loads the handler address (00000010H) of the non-maskable interrupt routine to the PC, and transfers control.

The servicing of a non-maskable interrupt is shown below.

Figure 21-1. Non-Maskable Interrupt Servicing

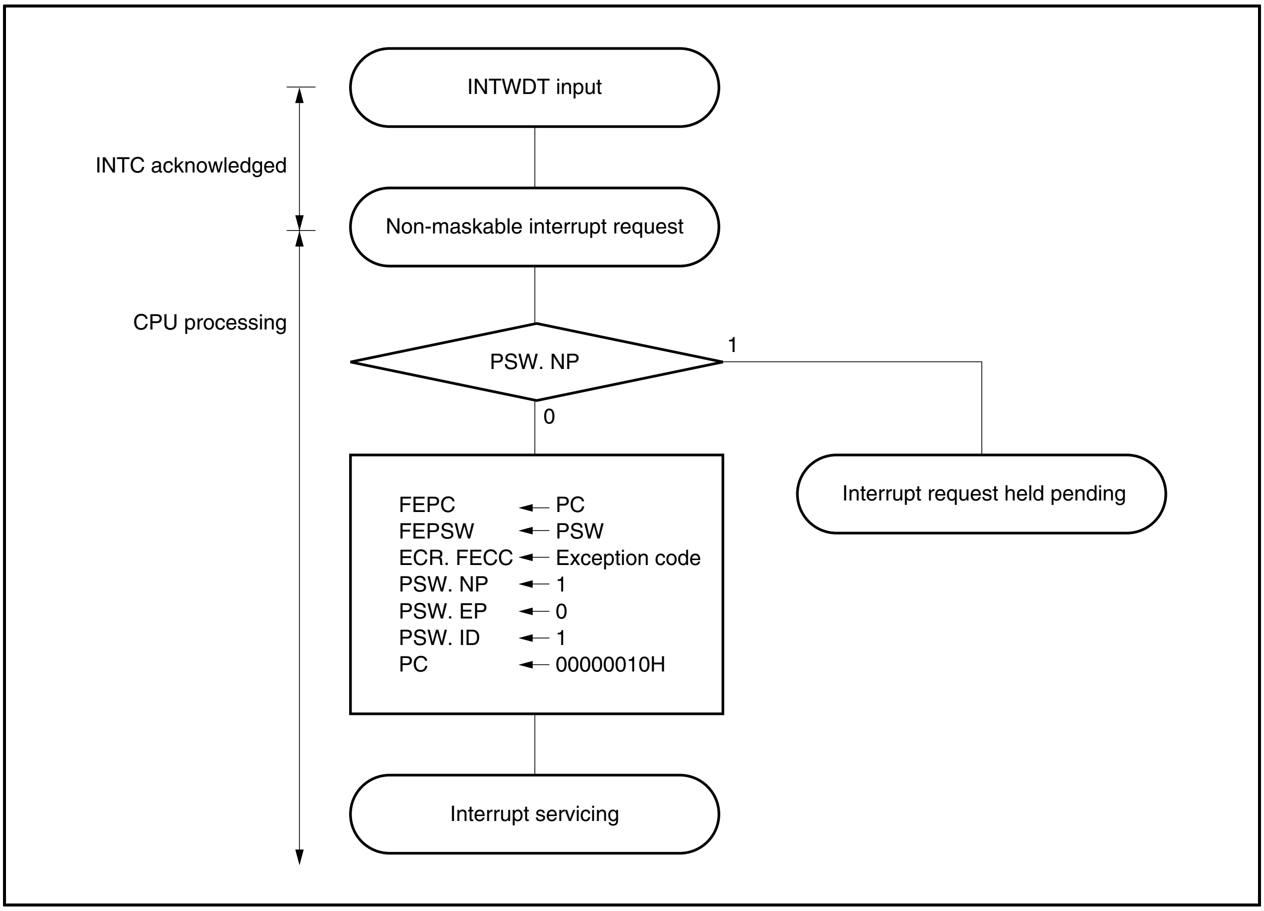
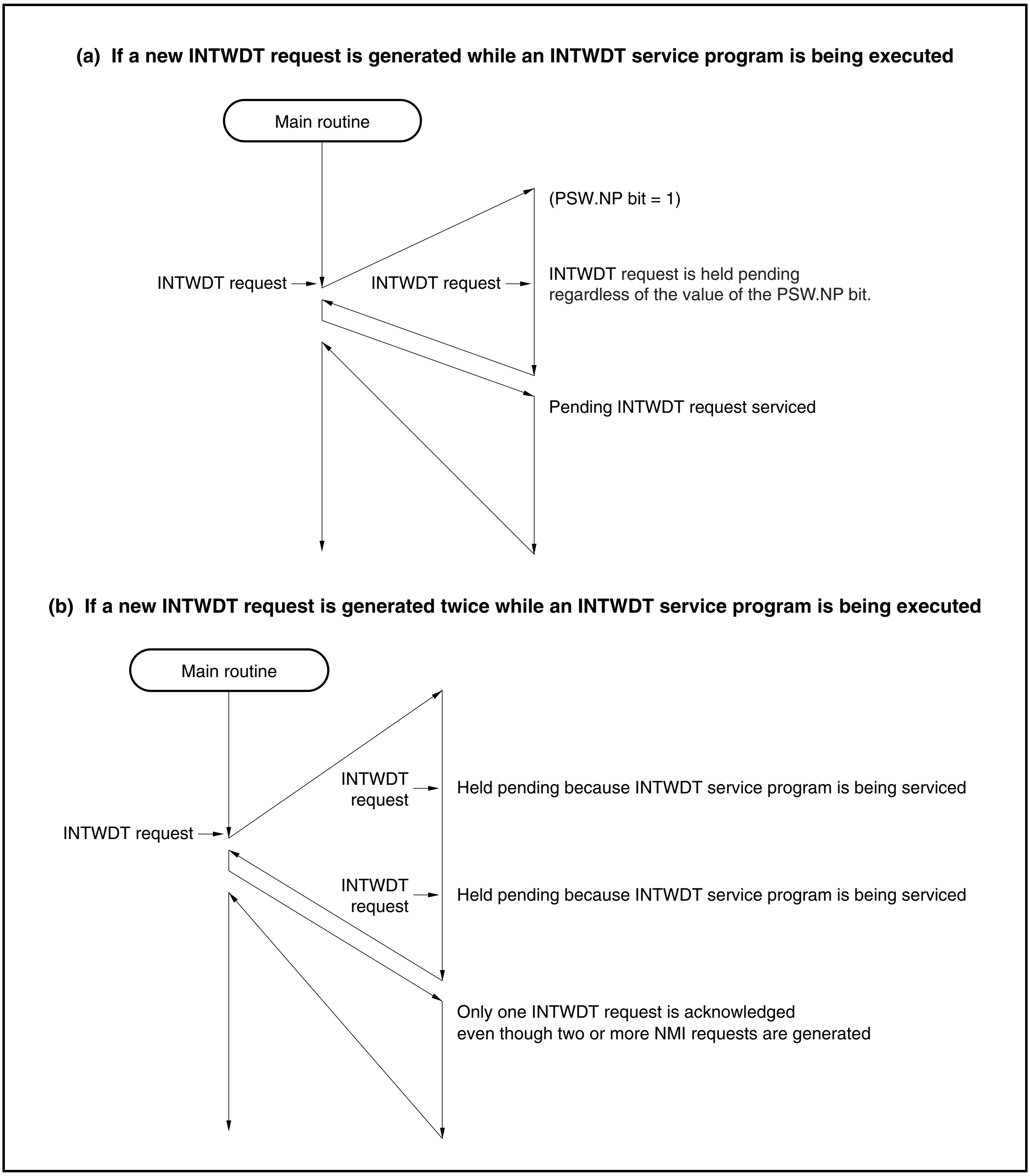


Figure 21-2. Acknowledging Non-Maskable Interrupt Request


(a) If a new INTWDT request is generated while an INTWDT service program is being executed
Main routine
(PSW.NP bit = 1)
INTWDT request
INTWDT request
INTWDT request is held pending regardless of the value of the PSW.NP bit.
Pending INTWDT request serviced
(b) If a new INTWDT request is generated twice while an INTWDT service program is being executed
Main routine
INTWDT request
Held pending because INTWDT service program is being serviced
INTWDT request
INTWDT request
Held pending because INTWDT service program is being serviced
Only one INTWDT request is acknowledged even though two or more NMI requests are generated

21.2.2 Return processing

Execution is returned from non-maskable interrupt servicing by using the RETI instruction.

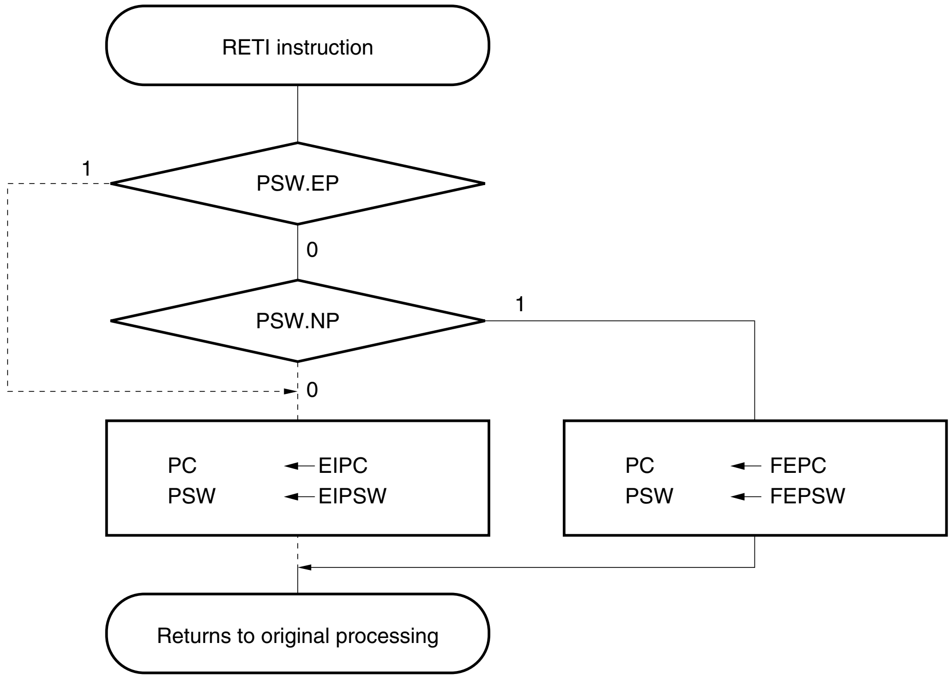
When the RETI instruction is executed, the CPU performs the following processing and transfers control to the address of the return PC.

<1> Loads the saved PC and PSW from FEPC and FEPSW because the PSW.EP bit is 0 and the PSW.NP bit is 1.

<2> Transfers control back to the address of the return PC and PSW.

The following illustrates how the RETI instruction is processed.

Figure 21-3. RETI Instruction Processing



Caution **When the EP and NP bits are changed by the LDSR instruction during non-maskable interrupt servicing, to restore the PC and PSW correctly when returning by using the RETI instruction, the EP bit must be cleared (= 0) and the NP bit must be set (= 1) using the LDSR instruction immediately before the RETI instruction.**

Remark The solid line shows the CPU processing flow.

21.2.3 Non-maskable interrupt status flag (NP)

The NP flag is a status flag that indicates that a non-maskable interrupt (INTWDT) is being serviced. The NP flag is allocated to the PSW.

This flag is set when an INTWDT interrupt request signal has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.

The flag is cleared to 00000020H after reset.

After reset: 00000020H

	31 to 8	7	6	5	4	3	2	1	0
PSW	0	**NP**	EP	ID	SAT	CY	OV	S	Z

NP	Non-maskable interrupt (INTWDT) servicing status
0	No non-maskable interrupt servicing
1	Non-maskable interrupt servicing in progress

21.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850E/IG4-H and V850E/IH4-H have 105 maskable interrupt sources.

If two or more maskable interrupt request signals are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request signal has been acknowledged, interrupts are disabled (DI) and the subsequent maskable interrupt request signals are not acknowledged.

When the EI instruction is executed in an interrupt service routine, interrupts are enabled (EI), which enables servicing of interrupts having a higher priority than that of the interrupt request signal currently being serviced (specified by the interrupt control register). Interrupts with the same priority level cannot be nested.

To enable multiple interrupt servicing, however, save EIPC and EIPSW to memory or registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

21.3.1 Operation

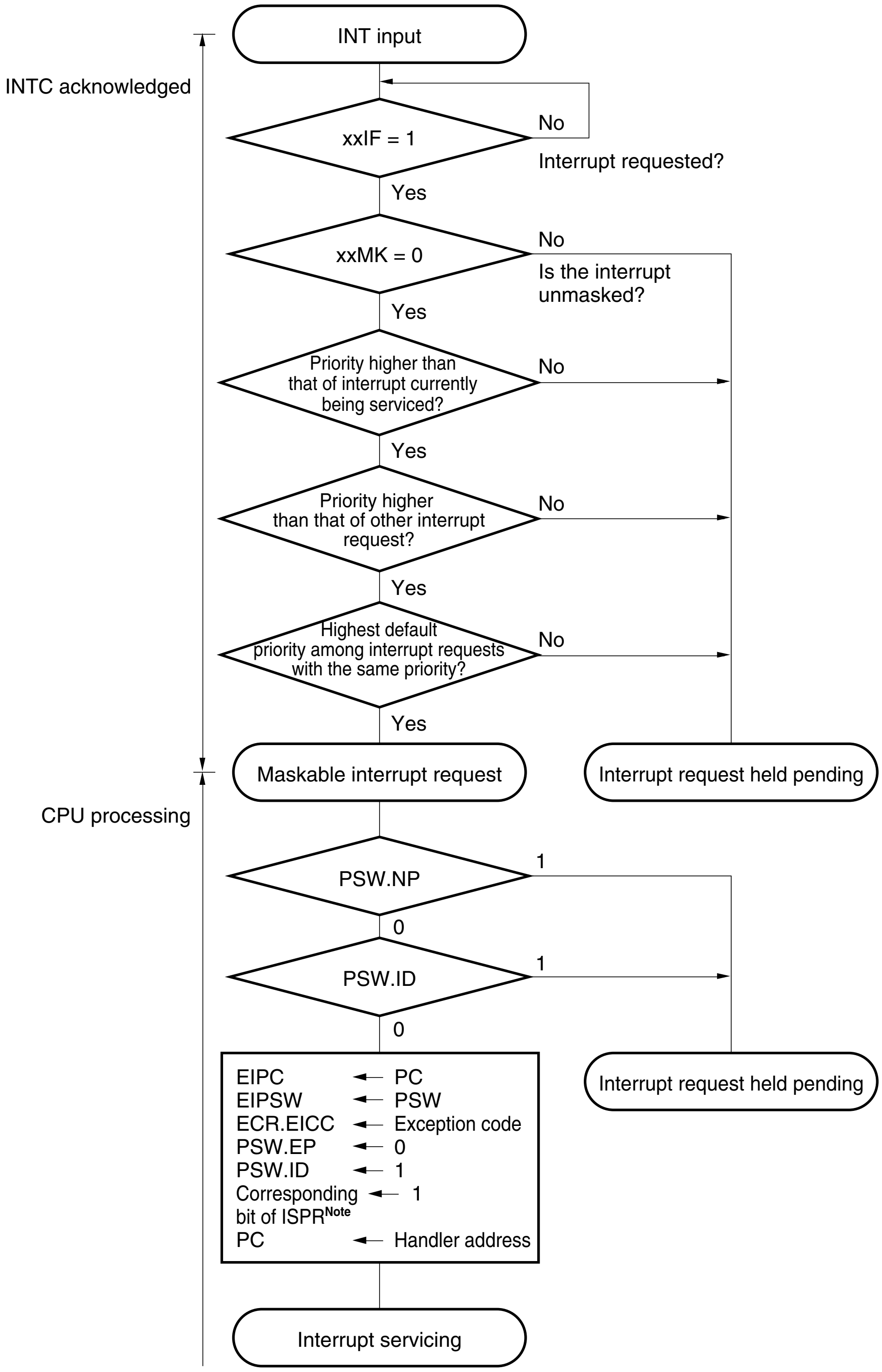
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the current PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the PSW.ID bit to 1 and clears the PSW.EP bit to 0.
<5> Sets the handler address corresponding to each interrupt to the PC, and transfers control.

A maskable interrupt request signal masked by interrupt controller (INTC) and a maskable interrupt request signal generated while another interrupt is being serviced (while PSW.NP bit is 1 or ID bit is 1) are held pending in the INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request signal if either the maskable interrupt is unmasked or the NP and ID bits are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 21-4. Maskable Interrupt Servicing


INT input
INTC acknowledged
xxIF = 1
No
Interrupt requested?
Yes
xxMK = 0
No
Is the interrupt unmasked?
Yes
Priority higher than that of interrupt currently being serviced?
No
Yes
Priority higher than that of other interrupt request?
No
Yes
Highest default priority among interrupt requests with the same priority?
No
Yes
Maskable interrupt request
Interrupt request held pending
CPU processing
PSW.NP
1
0
PSW.ID
1
0
EIPC ← PC
EIPSW ← PSW
ECR.EICC ← Exception code
PSW.EP ← 0
PSW.ID ← 1
Corresponding bit of ISPRNote ← 1
PC ← Handler address
Interrupt request held pending
Interrupt servicing

Note For details of the ISPR register, see **21.3.6 In-service priority register (ISPR)**.

21.3.2 Return processing

Execution is returned from non-maskable interrupt servicing by using the RETI instruction.

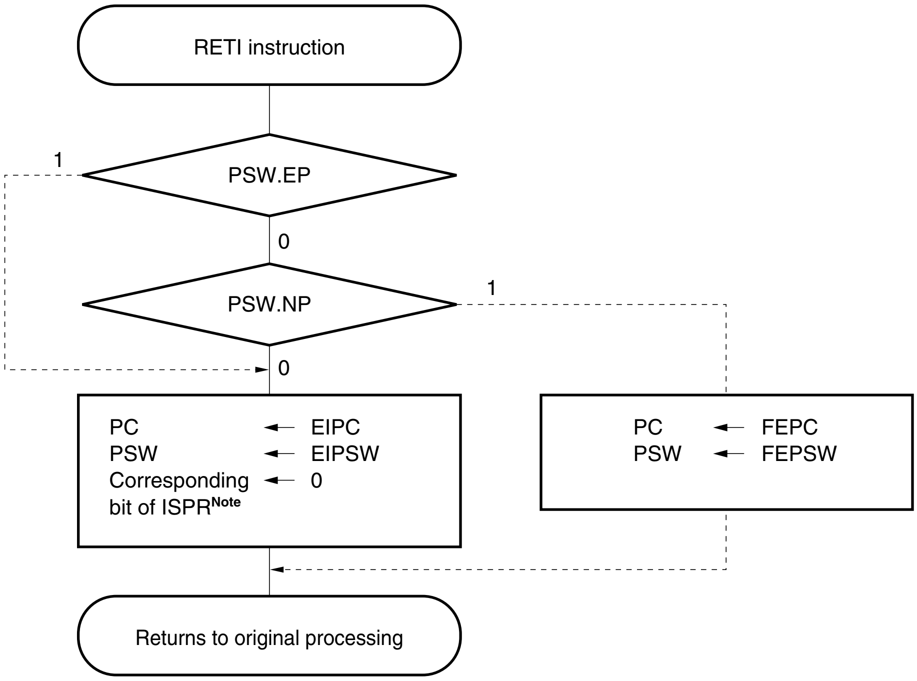
When the RETI instruction is executed, the CPU performs the following processing and transfers control to the address of the return PC.

<1> Loads the values of the PC and the PSW from EIPC and EIPSW, respectively, because the PSW.EP bit is 0 and the PSW.NP bit is 0.

<2> Transfers control back to the address of the return PC and PSW.

The processing of the RETI instruction is shown below.

Figure 21-5. RETI Instruction Processing



Note For the ISPR register, see **21.3.6 In-service priority register (ISPR)**.

Caution When the EP and NP bits are changed by the LDSR instruction during non-maskable interrupt servicing, to restore the PC and PSW correctly when returning by using the RETI instruction, the EP bit must be cleared (= 0) and the NP bit must be set (= 1) using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

21.3.3 Priorities of maskable interrupts

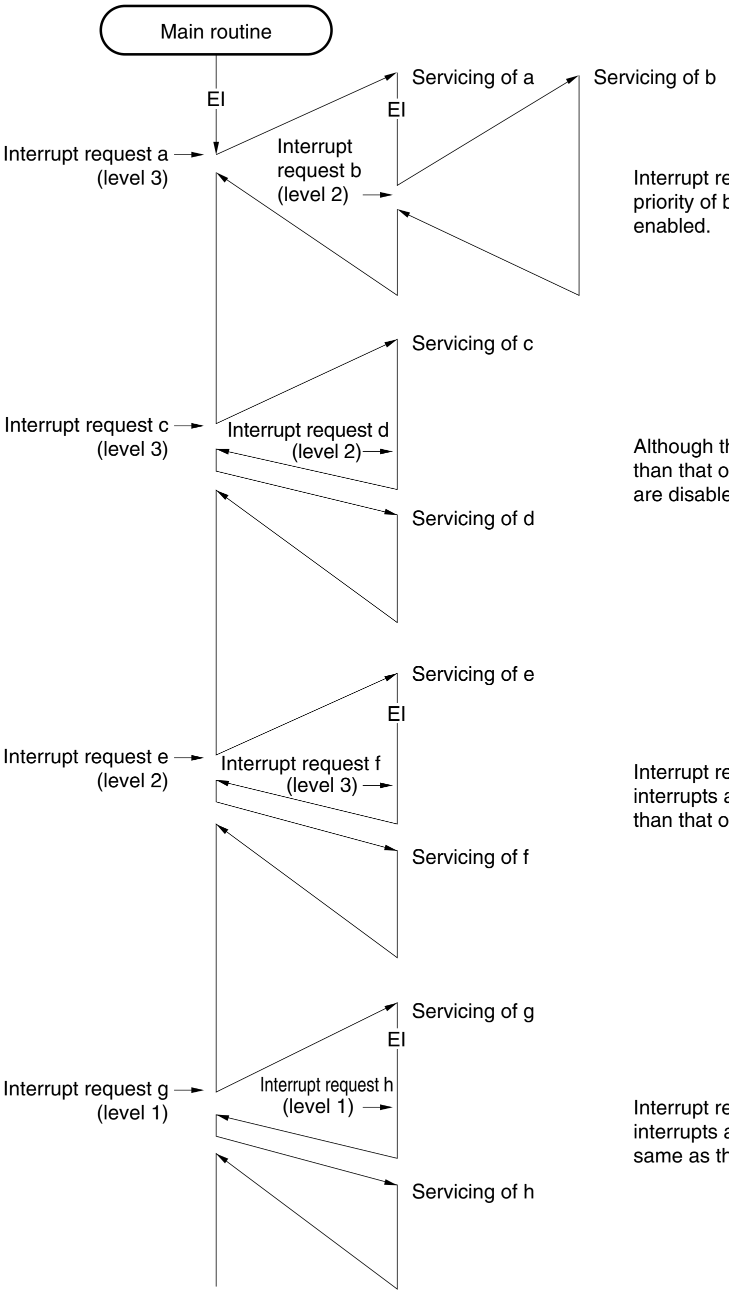
The INTC provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit occur at the same time, the interrupts are serviced according to the priority levels assigned to the corresponding interrupt requests (default priority level) beforehand. For more information, see **Table 21-1 Interrupt Source List**. Programmable priority control customizes interrupt request signals into eight levels according to the setting of the priority level specification flag.

Note that when an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt servicing program) to set the interrupt enabled mode.

Remark xx: Identification name of each peripheral unit (see **Table 21-2**)
n: Peripheral unit number (see **Table 21-2**)

Figure 21-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)



Interrupt request b is acknowledged because the priority of b is higher than that of a and interrupts are enabled.

Although the priority of interrupt request d is higher than that of c, d is held pending because interrupts are disabled.

Interrupt request f is held pending even though interrupts are enabled because its priority is lower than that of e.

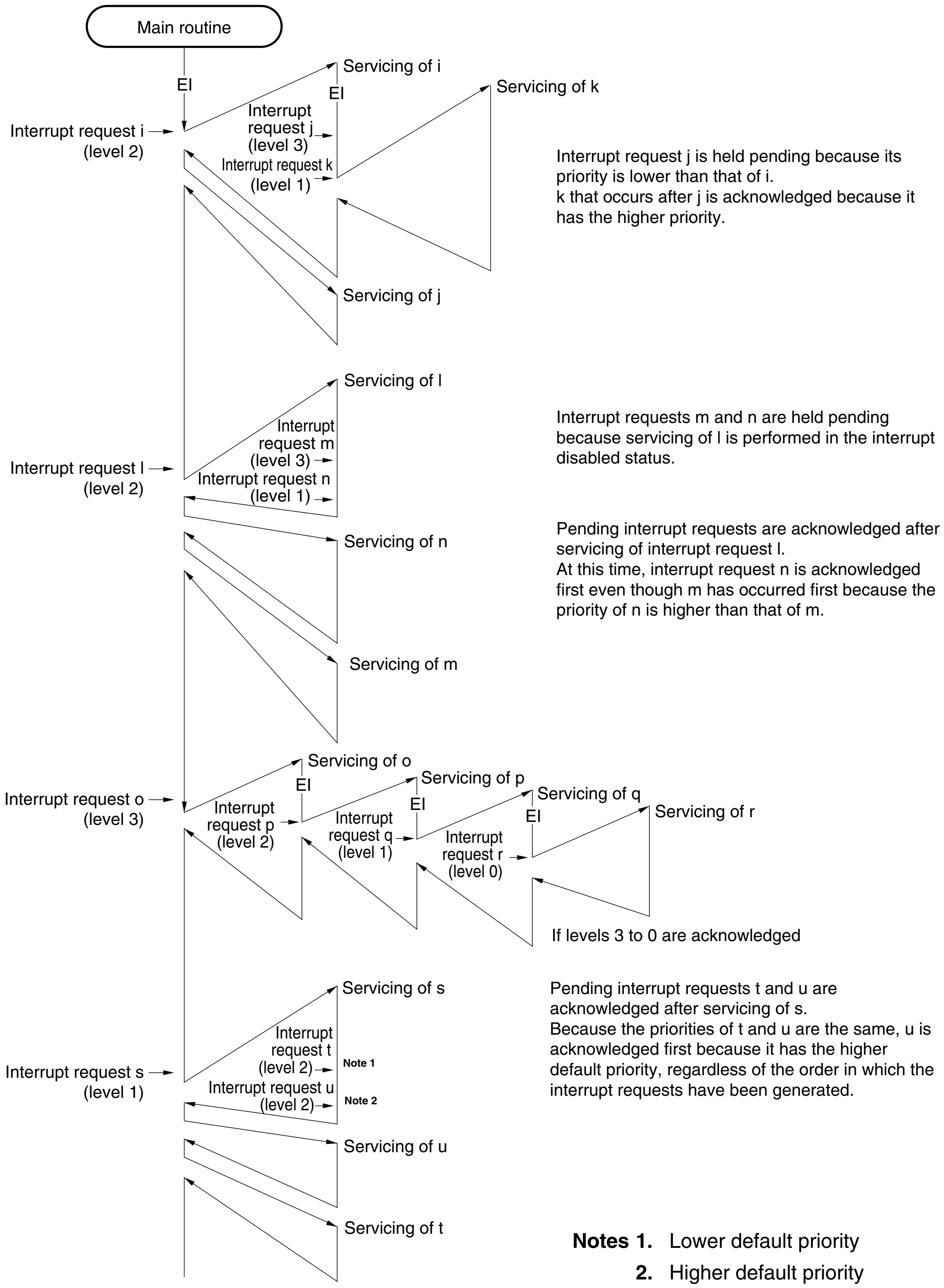
Interrupt request h is held pending even though interrupts are enabled because its priority is the same as that of g.

Caution To service multiple interrupts, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to u in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 21-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (2/2)


Main routine
EI
Interrupt request i (level 2)
Servicing of i
EI
Interrupt request j (level 3)
Interrupt request k (level 1)
Servicing of k
Servicing of j
Servicing of l
Interrupt request l (level 2)
Interrupt request m (level 3)
Interrupt request n (level 1)
Servicing of n
Servicing of m
Interrupt request o (level 3)
Servicing of o
EI
Interrupt request p (level 2)
Servicing of p
EI
Interrupt request q (level 1)
Servicing of q
EI
Interrupt request r (level 0)
Servicing of r
If levels 3 to 0 are acknowledged
Servicing of s
Interrupt request s (level 1)
Interrupt request t (level 2)
Note 1
Interrupt request u (level 2)
Note 2
Servicing of u
Servicing of t

Interrupt request j is held pending because its priority is lower than that of i.
k that occurs after j is acknowledged because it has the higher priority.

Interrupt requests m and n are held pending because servicing of l is performed in the interrupt disabled status.

Pending interrupt requests are acknowledged after servicing of interrupt request l.
At this time, interrupt request n is acknowledged first even though m has occurred first because the priority of n is higher than that of m.

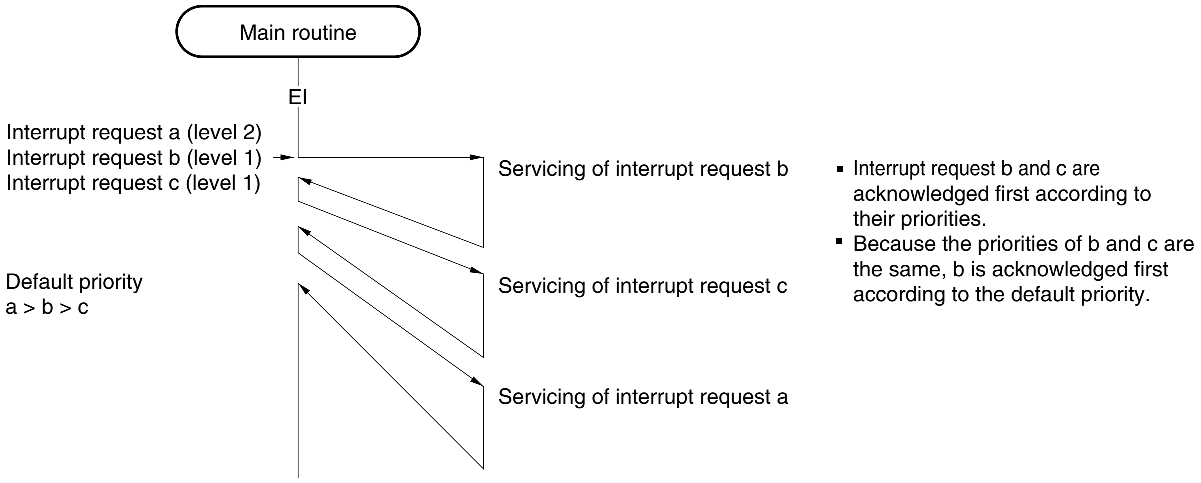
Pending interrupt requests t and u are acknowledged after servicing of s.
Because the priorities of t and u are the same, u is acknowledged first because it has the higher default priority, regardless of the order in which the interrupt requests have been generated.

Notes 1. Lower default priority

2. Higher default priority

Caution To service multiple interrupts, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Figure 21-7. Example of Servicing Interrupt Requests Generated Simultaneously


Main routine
EI
Interrupt request a (level 2)
Interrupt request b (level 1)
Interrupt request c (level 1)
Default priority
a > b > c
Servicing of interrupt request b
Servicing of interrupt request c
Servicing of interrupt request a
▪ Interrupt request b and c are acknowledged first according to their priorities.
▪ Because the priorities of b and c are the same, b is acknowledged first according to the default priority.

Caution **To service multiple interrupts, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.**

Remarks 1. a to c in the figure are assumed names given to interrupt request signals for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt request signals.

21.3.4 Interrupt control registers (xxICn)

An xxICn register is assigned to each interrupt request signal (maskable interrupt) and sets the control conditions for each maskable interrupt request.

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 47H.

Cautions 1. Disable interrupts (DI) before reading the xxICn.xxIFn bit. If the xxIFn bit is read while interrupts are enabled (EI), the correct value may not be read if acknowledging an interrupt and reading the bit conflict.

2. When manipulating the xxICn.xxMKn bit while interrupt requests may occur (including the state in which interrupts are disabled (DI)), be sure to use a bit manipulation instruction or use the IMRm.xxMKn bit (m = 0 to 6).

After reset: 47H R/W Address: FFFFF110H to FFFFF1E0H

	<7>	<6>	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	Interrupt request flag[Note]
0	Interrupt request not issued
1	Interrupt request issued

xxMKn	Interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled (pending)

xxPRn2	xxPRn1	xxPRn0	Interrupt priority specification bit
0	0	0	Specifies level 0 (highest).
0	0	1	Specifies level 1.
0	1	0	Specifies level 2.
0	1	1	Specifies level 3.
1	0	0	Specifies level 4.
1	0	1	Specifies level 5.
1	1	0	Specifies level 6.
1	1	1	Specifies level 7 (lowest).

Note The flag xxIFn is reset automatically by the hardware if an interrupt request signal is acknowledged.

Remark xx: Identification name of each peripheral unit (see **Table 21-2**)
n: Peripheral unit number (see **Table 21-2**)

The addresses and bits of the interrupt control registers are as follows.

Table 21-2. Addresses and Bits of Interrupt Control Registers (1/3)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	LVILIC	LVILIF	LVILMK	0	0	0	LVILPR2	LVILPR1	LVILPR0
FFFFF112H	LVIHIC	LVIHIF	LVIHMK	0	0	0	LVIHPR2	LVIHPR1	LVIHPR0
FFFFF114H	PIC00	PIF00	PMK00	0	0	0	PPR002	PPR001	PPR000
FFFFF116H	PIC01	PIF01	PMK01	0	0	0	PPR012	PPR011	PPR010
FFFFF118H	PIC02	PIF02	PMK02	0	0	0	PPR022	PPR021	PPR020
FFFFF11AH	PIC03	PIF03	PMK03	0	0	0	PPR032	PPR031	PPR030
FFFFF11CH	PIC04	PIF04	PMK04	0	0	0	PPR042	PPR041	PPR040
FFFFF11EH	PIC05	PIF05	PMK05	0	0	0	PPR052	PPR051	PPR050
FFFFF120H	PIC06	PIF06	PMK06	0	0	0	PPR062	PPR061	PPR060
FFFFF122H	PIC07	PIF07	PMK07	0	0	0	PPR072	PPR071	PPR070
FFFFF124H	PIC08	PIF08	PMK08	0	0	0	PPR082	PPR081	PPR080
FFFFF126H	PIC09	PIF09	PMK09	0	0	0	PPR092	PPR091	PPR090
FFFFF128H	PIC10	PIF10	PMK10	0	0	0	PPR102	PPR101	PPR100
FFFFF12AH	PIC11	PIF11	PMK11	0	0	0	PPR112	PPR111	PPR110
FFFFF12CH	PIC12	PIF12	PMK12	0	0	0	PPR122	PPR121	PPR120
FFFFF12EH	PIC13	PIF13	PMK13	0	0	0	PPR132	PPR131	PPR130
FFFFF130H	PIC14	PIF14	PMK14	0	0	0	PPR142	PPR141	PPR140
FFFFF132H	PIC15	PIF15	PMK15	0	0	0	PPR152	PPR151	PPR150
FFFFF134H	PIC16	PIF16	PMK16	0	0	0	PPR162	PPR161	PPR160
FFFFF136H	PIC17	PIF17	PMK17	0	0	0	PPR172	PPR171	PPR170
FFFFF138H	PIC18	PIF18	PMK18	0	0	0	PPR182	PPR181	PPR180
FFFFF13AH	PIC19	PIF19	PMK19	0	0	0	PPR192	PPR191	PPR190
FFFFF13CH	CMPIC0L	CMPIF0L	CMPMK0L	0	0	0	CMPPR0L2	CMPPR0L1	CMPPR0L0
FFFFF13EH	CMPIC0F	CMPIF0F	CMPMK0F	0	0	0	CMPPR0F2	CMPPR0F1	CMPPR0F0
FFFFF140H	CMPIC1L	CMPIF1L	CMPMK1L	0	0	0	CMPPR1L2	CMPPR1L1	CMPPR1L0
FFFFF142H	CMPIC1F	CMPIF1F	CMPMK1F	0	0	0	CMPPR1F2	CMPPR1F1	CMPPR1F0
FFFFF144H	TB0OVIC	TB0OVIF	TB0OVMK	0	0	0	TB0OVPR2	TB0OVPR1	TB0OVPR0
FFFFF146H	TB0CCIC0	TB0CCIF0	TB0CCMK0	0	0	0	TB0CCPR02	TB0CCPR01	TB0CCPR00
FFFFF148H	TB0CCIC1	TB0CCIF1	TB0CCMK1	0	0	0	TB0CCPR12	TB0CCPR11	TB0CCPR10
FFFFF14AH	TB0CCIC2	TB0CCIF2	TB0CCMK2	0	0	0	TB0CCPR22	TB0CCPR21	TB0CCPR20
FFFFF14CH	TB0CCIC3	TB0CCIF3	TB0CCMK3	0	0	0	TB0CCPR32	TB0CCPR31	TB0CCPR30
FFFFF14EH	TB1OVIC	TB1OVIF	TB1OVMK	0	0	0	TB1OVPR2	TB1OVPR1	TB1OVPR0
FFFFF150H	TB1CCIC0	TB1CCIF0	TB1CCMK0	0	0	0	TB1CCPR02	TB1CCPR01	TB1CCPR00
FFFFF152H	TB1CCIC1	TB1CCIF1	TB1CCMK1	0	0	0	TB1CCPR12	TB1CCPR11	TB1CCPR10
FFFFF154H	TB1CCIC2	TB1CCIF2	TB1CCMK2	0	0	0	TB1CCPR22	TB1CCPR21	TB1CCPR20
FFFFF156H	TB1CCIC3	TB1CCIF3	TT0CCMK3	0	0	0	TB1CCPR32	TB1CCPR31	TB1CCPR30
FFFFF158H	TT0OVIC	TT0OVIF	TT0OVMK	0	0	0	TT0OVPR2	TT0OVPR1	TT0OVPR0
FFFFF15AH	TT0CCIC0	TT0CCIF0	TT0CCMK0	0	0	0	TT0CCPR02	TT0CCPR01	TT0CCPR00
FFFFF15CH	TT0CCIC1	TT0CCIF1	TT0CCMK1	0	0	0	TT0CCPR12	TT0CCPR11	TT0CCPR10
FFFFF15EH	TT0IECIC	TT0IECIF	TT0IECMK	0	0	0	TT0IECPR2	TT0IECPR1	TT0IECPR0
FFFFF160H	TT1OVIC	TT1OVIF	TT1OVMK	0	0	0	TT1OVPR2	TT1OVPR1	TT1OVPRO
FFFFF162H	TT1CCIC0	TT1CCIF0	TT1CCMK0	0	0	0	TT1CCPR02	TT1CCPR01	TT1CCPR00
FFFFF164H	TT1CCIC1	TT1CCIF1	TT1CCMK1	0	0	0	TT1CCPR12	TT1CCPR11	TT1CCPR10

Table 21-2. Addresses and Bits of Interrupt Control Registers (2/3)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF166H	TT1IECIC	TT1IECIF	TT1IECMK	0	0	0	TT1IECPR2	TT1IECPR1	TT1IECPR0
FFFFF168H	TT2OVIC	TT2OVIF	TT2OVMK	0	0	0	TT2OVPR2	TT2OVPR1	TT2OVPR0
FFFFF16AH	TT2CCIC0	TT2CCIF0	TT2CCMK0	0	0	0	TT2CCPR02	TT2CCPR01	TT2CCPR00
FFFFF16CH	TT2CCIC1	TT2CCIF1	TT2CCMK1	0	0	0	TT2CCPR12	TT2CCPR11	TT2CCPR10
FFFFF16EH	TT3OVIC	TT3OVIF	TT3OVMK	0	0	0	TT3OVPR2	TT3OVPR1	TT3OVPR0
FFFFF170H	TT3CCIC0	TT3CCIF0	TT3CCMK0	0	0	0	TT3CCPR02	TT3CCPR01	TT3CCPR00
FFFFF172H	TT3CCIC1	TT3CCIF1	TT3CCMK1	0	0	0	TT3CCPR12	TT3CCPR11	TT3CCPR10
FFFFF174H	TA0OVIC	TA0OVIF	TA0OVMK	0	0	0	TA0OVPR2	TA0OVPR1	TA0OVPR0
FFFFF176H	TA0CCIC0	TA0CCIF0	TA0CCMK0	0	0	0	TA0CCPR02	TA0CCPR01	TA0CCPR00
FFFFF178H	TA0CCIC1	TA0CCIF1	TA0CCMK1	0	0	0	TA0CCPR12	TA0CCPR11	TA0CCPR10
FFFFF17AH	TA1OVIC	TA1OVIF	TA1OVMK	0	0	0	TA1OVPR2	TA1OVPR1	TA1OVPR0
FFFFF17CH	TA1CCIC0	TA1CCIF0	TA1CCMK0	0	0	0	TA1CCPR02	TA1CCPR01	TA1CCPR00
FFFFF17EH	TA1CCIC1	TA1CCIF1	TA1CCMK1	0	0	0	TA1CCPR12	TA1CCPR11	TA1CCPR10
FFFFF180H	TA2OVIC	TA2OVIF	TA2OVMK	0	0	0	TA2OVPR2	TA2OVPR1	TA2OVPR0
FFFFF182H	TA2CCIC0	TA2CCIF0	TA2CCMK0	0	0	0	TA2CCPR02	TA2CCPR01	TA2CCPR00
FFFFF184H	TA2CCIC1	TA2CCIF1	TA2CCMK1	0	0	0	TA2CCPR12	TA2CCPR11	TA2CCPR10
FFFFF186H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF188H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF18AH	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF18CH	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF18EH	DMAIC4	DMAIF4	DMAMK4	0	0	0	DMAPR42	DMAPR41	DMAPR40
FFFFF190H	DMAIC5	DMAIF5	DMAMK5	0	0	0	DMAPR52	DMAPR51	DMAPR50
FFFFF192H	UREIC	UREIF	UREMK	0	0	0	UREPR2	UREPR1	UREPR0
FFFFF194H	URIC	URIF	URMK	0	0	0	URPR2	URPR1	URPR0
FFFFF196H	UTIC	UTIF	UTMK	0	0	0	UTPR2	UTPR1	UTPR0
FFFFF198H	UIFIC	UIFIF	UIFMK	0	0	0	UIFPR2	UIFPR1	UIFPR0
FFFFF19AH	UTOIC	UTOIF	UTOMK	0	0	0	UTOPR2	UTOPR1	UTOPR0
FFFFF19CH	UA0REIC	UA0REIF	UA0REMK	0	0	0	UA0REPR2	UA0REPR1	UA0REPR0
FFFFF19EH	UA0RIC	UA0RIF	UA0RMK	0	0	0	UA0RPR2	UA0RPR1	UA0RPR0
FFFFF1A0H	UA0TIC	UA0TIF	UA0TMK	0	0	0	UA0TPR2	UA0TPR1	UA0TPR0
FFFFF1A2H	CF0REIC	CF0REIF	CF0REMK	0	0	0	CF0REPR2	CF0REPR1	CF0REPR0
FFFFF1A4H	CF0RIC	CF0RIF	CF0RMK	0	0	0	CF0RPR2	CF0RPR1	CF0RPR0
FFFFF1A6H	CF0TIC	CF0TIF	CF0TMK	0	0	0	CF0TPR2	CF0TPR1	CF0TPR0
FFFFF1A8H	UA1REIC	UA1REIF	UA1REMK	0	0	0	UA1REPR2	UA1REPR1	UA1REPR0
FFFFF1AAH	UA1RIC	UA1RIF	UA1RMK	0	0	0	UA1RPR2	UA1RPR1	UA1RPR0
FFFFF1ACH	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF1AEH	CF1REIC	CF1REIF	CF1REMK	0	0	0	CF1REPR2	CF1REPR1	CF1REPR0
FFFFF1B0H	CF1RIC	CF1RIF	CF1RMK	0	0	0	CF1RPR2	CF1RPR1	CF1RPR0
FFFFF1B2H	CF1TIC	CF1TIF	CF1TMK	0	0	0	CF1TPR2	CF1TPR1	CF1TPR0
FFFFF1B4H	UA2REIC	UA2REIF	UA2REMK	0	0	0	UA2REPR2	UA2REPR1	UA2REPR0
FFFFF1B6H	UA2RIC	UA2RIF	UA2RMK	0	0	0	UA2RPR2	UA2RPR1	UA2RPR0
FFFFF1B8H	UA2TIC	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0
FFFFF1BAH	CF2REIC	CF2REIF	CF2REMK	0	0	0	CF2REPR2	CF2REPR1	CF2REPR0

Table 21-2. Addresses and Bits of Interrupt Control Registers (3/3)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF1BCH	CF2RIC	CF2RIF	CF2RMK	0	0	0	CF2RPR2	CF2RPR1	CF2RPR0
FFFFF1BEH	CF2TIC	CF2TIF	CF2TMK	0	0	0	CF2TPR2	CF2TPR1	CF2TPR0
FFFFF1C0H	IICIC	IICIF	IICMK	0	0	0	IICPR2	IICPR1	IICPR0
FFFFF1C2H	AD0IC	AD0IF	AD0MK	0	0	0	AD0PR2	AD0PR1	AD0PR0
FFFFF1C4H	AD1IC	AD1IF	AD1MK	0	0	0	AD1PR2	AD1PR1	AD1PR0
FFFFF1C6H	AD2IC	AD2IF	AD2MK	0	0	0	AD2PR2	AD2PR1	AD2PR0
FFFFF1C8H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF1CAH	TM1EQIC0	TM1EQIF0	TM1EQMK0	0	0	0	TM1EQPR02	TM1EQPR01	TM1EQPR00
FFFFF1CCH	TM2EQIC0	TM2EQIF0	TM2EQMK0	0	0	0	TM2EQPR02	TM2EQPR01	TM2EQPR00
FFFFF1CEH	TM3EQIC0	TM3EQIF0	TM3EQMK0	0	0	0	TM3EQPR02	TM3EQPR01	TM3EQPR00
FFFFF1D0H	ADT0IC	ADT0IF	ADT0MK	0	0	0	ADT0PR2	ADT0PR1	ADT0PR0
FFFFF1D2H	ADT1IC	ADT1IF	ADT1MK	0	0	0	ADT1PR2	ADT1PR1	ADT1PR0
FFFFF1D4H	UFIC0	UFIF0	UFMK0	0	0	0	UFPR02	UFPR01	UFPR00
FFFFF1D6H	UFIC1	UFIF1	UFMK1	0	0	0	UFPR12	UFPR11	UFPR10
FFFFF1D8H	DMAIC6	DMAIF6	DMAMK6	0	0	0	DMAPR62	DMAPR61	DMAPR60
FFFFF1DAH	TB0OVBIC	TB0OVBIF	TB0OVBMK	0	0	0	TB0OVBPR2	TB0OVBPR1	TB0OVBPR0
FFFFF1DCH	TB0CCBIC0	TB0CCBIF0	TB0CCBMK0	0	0	0	TB0OVBPR02	TB0OVBPR01	TB0OVBPR00
FFFFF1DEH	TB1OVBIC	TB1OVBIF	TB1OVBMK	0	0	0	TB1OVBPR2	TB1OVBPR1	TB1OVBPR0
FFFFF1E0H	TB1CCBIC0	TB1CCBIF0	TB1CCBMK0	0	0	0	TB1CCBPR02	TB1CCBPR01	TB1CCBPR00

21.3.5 Interrupt mask registers 0 to 6 (IMR0 to IMR6)

The IMR0 to IMR6 registers specify masking of the maskable interrupts. The IMR0.xxMKn to IMR6.xxMKn bits are equivalent to the xxICn.xxMKn bit.

Each IMRm register can be read or written in 16-bit units (m = 0 to 6).

If the higher 8 bits of each IMRm register are used as the IMRmH register and the lower 8 bits as the IMRmL register, these registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to FFFFH.

Caution The device file defines the xxICn.xxMKn bit as a reserved word. If a bit is manipulated using the name of xxMKn, the values of the xxICn register, instead of the IMRm register, are rewritten (as a result, the values of the IMRm register are also rewritten).

(1/2)

After reset: FFFFH R/W Address: IMR6 FFFFF10CH
IMR6L FFFFF10CH, IMR6H FFFFF10DH

	15	14	13	12	11	10	9	8
IMR6 (IMR6H[Note])	1	1	1	1	1	1	1	TB1CCBMK0
	7	6	5	4	3	2	1	0
(IMR6L)	TB1OVBMK	TB0CCBMK0	TB0OVBMK	DMAMK6	UFMK1	UFMK0	ADT1MK	ADT0MK

After reset: FFFFH R/W Address: IMR5 FFFFF10AH
IMR5L FFFFF10AH, IMR5H FFFFF10BH

	15	14	13	12	11	10	9	8
IMR5 (IMR5H[Note])	TM3EQMK0	TM2EQMK0	TM1EQMK0	TM0EQMK0	AD2MK	AD1MK	AD0MK	IICMK
	7	6	5	4	3	2	1	0
(IMR5L)	CF2TMK	CF2RMK	CF2REMK	UA2TMK	UA2RMK	UA2REMK	CF1TMK	CF1RMK

After reset: FFFFH R/W Address: IMR4 FFFFF108H
IMR4L FFFFF108H, IMR4H FFFFF109H

	15	14	13	12	11	10	9	8
IMR4 (IMR4H[Note])	CF1REMK	UA1TMK	UA1RMK	UA1REMK	CF0TMK	CF0RMK	CF0REMK	UA0TMK
	7	6	5	4	3	2	1	0
(IMR4L)	UA0RMK	UA0REMK	UTOMK	UIFMK	UTMK	URMK	UREMK	DMAMK5

After reset: FFFFH R/W Address: IMR3 FFFFF106H
IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR2 (IMR3H[Note])	DMAMK4	DMAMK3	DMAMK2	DMAMK1	DMAMK0	TA2CCMK1	TA2CCMK0	TA2OVMK
	7	6	5	4	3	2	1	0
(IMR3L)	TA1CCMK1	TA1CCMK0	TA1OVMK	TA0CCMK1	TA0CCMK0	TA0OVMK	TT3CCMK1	TT3CCMK0

After reset: FFFFH R/W Address: IMR2 FFFFF104H
IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H[Note])	TT3OVMK	TT2CCMK1	TT2CCMK0	TT2OVMK	TT1ECMK	TT1CCMK1	TT1CCMK0	TT1OVMK
	7	6	5	4	3	2	1	0
(IMR2L)	TT0IECMK	TT0CCMK1	TT0CCMK0	TT0OVMK	TB1CCMK3	TB1CCMK2	TB1CCMK1	TB1CCMK0

After reset: FFFFH R/W Address: IMR1 FFFFF102H
IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR0 (IMR1H[Note])	TB1OVMK	TB0CCMK3	TB0CCMK2	TB0CCMK1	TB0CCMK0	TB0OVMK	CMPMK1F	CMPMK1L
	7	6	5	4	3	2	1	0
(IMR1L)	CMPMK0F	CMPMK0L	PMK19	PMK18	PMK17	PMK16	PMK15	PMK14

Note To read or write bits 15 to 8 of the IMR1 to IMR6 registers in 8-bit or 1-bit units, specify these bits as bits 7 to 0 of the IMR1H to IMR6H registers.

Caution Set bits 15 to 9 of the IMR6 register (bits 7 to 1 of the IMR6H register) to 1. The operation when these settings are changed is not guaranteed.

(2/2)

After reset: FFFFH R/W Address: IMR0 FFFFF100H
IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H[Note])	PMK13	PMK12	PMK11	PMK10	PMK09	PMK08	PMK07	PMK09
	7	6	5	4	3	2	1	0
(IMR0L)	PMK05	PMK04	PMK03	PMK02	PMK01	PMK00	LVIHMK	LVILMK

xxMKn	Interrupt mask flag setting
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note To read or write bits 15 to 8 of the IMR0 register in 8-bit or 1-bit units, specify these bits as bits 7 to 0 of the IMR0H register.

Remark xx: Identification name of each peripheral unit (see **Table 21-2**)
n: Peripheral unit number (see **Table 21-2**)

21.3.6 In-service priority register (ISPR)

The ISPR register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request signal is acknowledged, the bit of this register corresponding to the priority level of that interrupt signal request is set to 1 and remains set while the interrupt is serviced.

When the RETI instruction is executed, the bit corresponding to the interrupt request signal having the highest priority is automatically cleared to 0 by hardware. However, it is not cleared to 0 when execution is returned from non-maskable interrupt servicing or an exception.

This register is read-only, in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution In the interrupt enabled (EI) state, if an interrupt is acknowledged during the reading of the ISPR register, the value of the ISPR register may be read after the bit is set (1) by this interrupt acknowledgment. To read the value of the ISPR register properly before interrupt acknowledgment, read it in the interrupt disabled (DI) state.

After reset: 00H R Address: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Priority of interrupt currently being acknowledged
0	Interrupt request signal with priority n is not acknowledged
1	Interrupt request signal with priority n is being acknowledged

Remark n: 0 to 7 (priority level)

21.3.7 Maskable interrupt status flag (ID)

The ID flag stores information regarding enabling or disabling interrupt requests. The ID flag is assigned to the PSW.

Reset sets this flag to 00000020H.

After rest: 00000020H

	31 to 8	7	6	5	4	3	2	1	0
PSW	0	NP	EP	ID	SAT	CY	OV	S	Z

ID	Maskable interrupt servicing specification[Note]
0	Maskable interrupt request signal acknowledgment enabled
1	Maskable interrupt request signal acknowledgment disabled

Note Interrupt disable flag (ID) function

ID is set (1) by the DI instruction and cleared (0) by the EI instruction. Its value is also rewritten by the RETI instruction, or by an LDSR instruction that writes data to the PSW.

Non-maskable interrupt request signals and exceptions are acknowledged regardless of this flag. When a maskable interrupt request signal is acknowledged, the ID flag is automatically set (1) by hardware.

An interrupt request signal generated during the acknowledgment disabled period (ID flag = 1) can be acknowledged when the xxICn.xxIFn bit is set (1), and the ID flag is cleared (0).

21.4 External Interrupt Request Input Pins (INTP00 to INTP19, INTADT0, INTADT1)

21.4.1 Noise elimination

(1) Noise elimination of INTP00 to INTP19, INTADT0, and INTADT1 pins

The INTP00 to INTP19, INTADT0, and INTADT1 pins incorporate a noise eliminator that uses analog filter. Unless, therefore, the input level of each pin is held for a certain time, an edge cannot be detected. An edge is detected after a certain time has elapsed.

(2) Noise elimination of INTP00 to INTP02, and INTP17 to INTP19 pins

The INTP00 to INTP02 and INTP17 to INTP19 pins incorporate a digital noise eliminator.

The sampling clock that performs digital sampling can be selected by the INTNFCm.INTNFCm2 to INTNFCm.INTNFCm0 bits (m = 00 to 02, and 17 to 19).

The system clock stops in the IDLE and STOP modes, so the INTP14 to INTP16 pins cannot be used to cancel the IDLE and STOP modes.

21.4.2 Edge detection

The valid edge of the INTn pin can be selected by program (n = P00 to P19, ADT0, and ADT1). The edge that can be selected as the valid edge is one of the following.

- Rising edge
- Falling edge
- Both the rising and falling edges

The edge-detected INTn signal becomes an interrupt source.

The valid edge is specified by the INTR0 to INTR3, ADTR, INTF0 to INTF3, and ADTF registers.

(1) External interrupt rising edge specification register 0 (INTR0), external interrupt falling edge specification register 0 (INTF0)

The INTR0 and INTF0 registers are used to specify the trigger mode of the INTP03 to INTP10 pins. The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

Caution When switching from the external interrupt function (alternate function) to the port mode, an edge might be detected. Therefore, specify the port mode after setting the INTFn and INTRn bits to 00 (n = 03 to 10).

After reset: 00H R/W Address: FFFFFC20H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTR0	INTR10	INTR09	INTR08	INTR07	INTR06	INTR05	INTR04	INTR03

After reset: 00H R/W Address: FFFFFC00H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTF0	INTF10	INTF09	INTF08	INTF07	INTF06	INTF05	INTF04	INTF03

Remark For the valid edge specification, see **Table 21-3**.

Table 21-3. Valid Edge Specification of INTP03 to INTP10 Pins

INTFn	INTRn	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to set the INTFn and INTRn bits to 00 when these registers are not used for the INTPn pins.

Remark n = 03 to 10

(2) External interrupt rising edge specification register 1 (INTR1), external interrupt falling edge specification register 1 (INTF1)

The INTR1 and INTF1 registers are used to specify the trigger mode of the INTP11 to INTP16 pins. The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

Caution When switching from the external interrupt function (alternate function) to the port mode, an edge might be detected. Therefore, specify the port mode after setting the INTFn and INTRn bits to 00 (n = 11 to 16).

After reset: 00H R/W Address: FFFFFC22H

	7	6	<5>	<4>	<3>	<2>	<1>	<0>
INTR1	0	0	INTR16	INTR15	INTR14	INTR13	INTR12	INTR11

After reset: 00H R/W Address: FFFFFC02H

	7	6	<5>	<4>	<3>	<2>	<1>	<0>
INTF1	0	0	INTF16	INTF15	INTF14	INTF13	INTF12	INTF11

Remark For the valid edge specification, see **Table 21-4**.

Table 21-4. Valid Edge Specification of INTP11 to INTP16 Pins

INTFn	INTRn	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to set the INTFn and INTRn bits to 00 when these registers are not used for the INTPn pins.

Remark n = 11 to 16

(3) External interrupt rising edge specification register 2 (INTR2), external interrupt falling edge specification register 2 (INTF2)

The INTR2 and INTF2 registers are used to specify the trigger mode of the INTP00 to INTP02 pins. The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

Caution When switching from the external interrupt function (alternate function) to the port mode, an edge might be detected. Therefore, specify the port mode after setting the INTFn and INTRn bits to 00 (n = 00 to 02).

After reset: 00H R/W Address: FFFFFC24H

	7	6	5	4	3	<2>	<1>	<0>
INTR2	0	0	0	0	0	INTR02	INTR01	INTR00

After reset: 00H R/W Address: FFFFFC04H

	7	6	5	4	3	<2>	<1>	<0>
INTF2	0	0	0	0	0	INTF02	INTF01	INTF00

Remark For the valid edge specification, see **Table 21-5**.

Table 21-5. Valid Edge Specification of INTP00 to INTP02 Pins

INTFn	INTRn	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution When not using these pins as the INTPn pins, be sure to set the INTFn and INTRn bits to 00.

Remark n = 00 to 02

(4) External interrupt rising edge specification register 3 (INTR3), external interrupt falling edge specification register 3 (INTF3)

The INTR3 and INTF3 registers are used to specify the trigger mode of the INTP17 to INTP19 pins. The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

Caution When switching from the external interrupt function (alternate function) to the port mode, an edge might be detected. Therefore, specify the port mode after setting the INTFn and INTRn bits to 00 (n = 17 to 19).

After reset: 00H R/W Address: FFFFFC26H

	7	6	5	4	3	<2>	<1>	<0>
INTR3	0	0	0	0	0	INTR19	INTR18	INTR17

After reset: 00H R/W Address: FFFFFC06H

	7	6	5	4	3	<2>	<1>	<0>
INTF3	0	0	0	0	0	INTF19	INTF18	INTF17

Remark For the valid edge specification, see **Table 21-6**.

Table 21-6. Valid Edge Specification of INTP17 to INTP19 Pins

INTFn	INTRn	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution When not using these pins as the INTPn pins, be sure to set the INTFn and INTRn bits to 00.

Remark n = 17 to 19

(5) A/D trigger rising edge, falling edge specification registers (ADTR, ADTF)

The ADTR and ADTF registers are used to specify the trigger mode of the ADTRG0/INTADT0 and ADTRG1/INTADT1 pins. The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

Caution When switching from the external trigger input of A/D converter n (alternate function)/external interrupt function (alternate function) to the port mode, an edge might be detected. Therefore, specify the port mode after setting the ADTFn and ADTRn bits to 00.

After reset: 00H R/W Address: FFFFF2F2H

	7	6	5	4	3	2	1	0
ADTR	0	0	0	0	0	0	ADTR1	ADTR0

After reset: 00H R/W Address: FFFFF2F0H

	7	6	5	4	3	2	1	0
ADTF	0	0	0	0	0	0	ADTF1	ADTF0

Remark For the valid edge specification, see **Table 21-7**.

Table 21-7. Valid Edge Specification of ADTRG0/INTADT0 and ADTRG1/INTADT1 Pins

ADTFn	ADTRn	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to set the ADTFn and ADTRn bits to 00 when these registers are not used for the ADTRGn/INTADTn pins.

Remark n = 0, 1

21.5 Software Exception

A software exception occurs when the CPU executes the TRAP instruction, and can always be acknowledged.

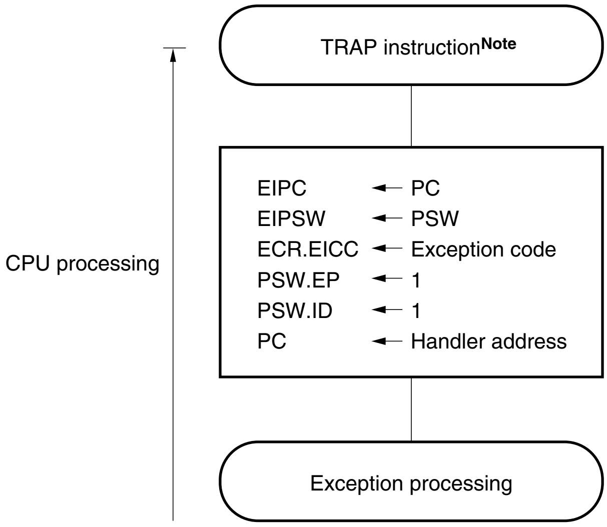
21.5.1 Operation

If a software exception occurs, the CPU performs the following processing and transfers control to the handler routine.

<1> Saves the current PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
<4> Sets the PSW.EP and PSW.ID bits (1).
<5> Sets the handler address (00000040H or 00000050H) for the software exception to the PC and transfers control.

The processing of a software exception is shown below.

Figure 21-8. Software Exception Processing



Note TRAP instruction format: TRAP vector (the vector is a value from 00H to 1FH.)

The handler address is determined by the TRAP instruction’s operand (vector). If the vector is 00H to 0FH, the handler address is 00000040H, and if the vector is 10H to 1FH, the handler address is 00000050H.

21.5.2 Return processing

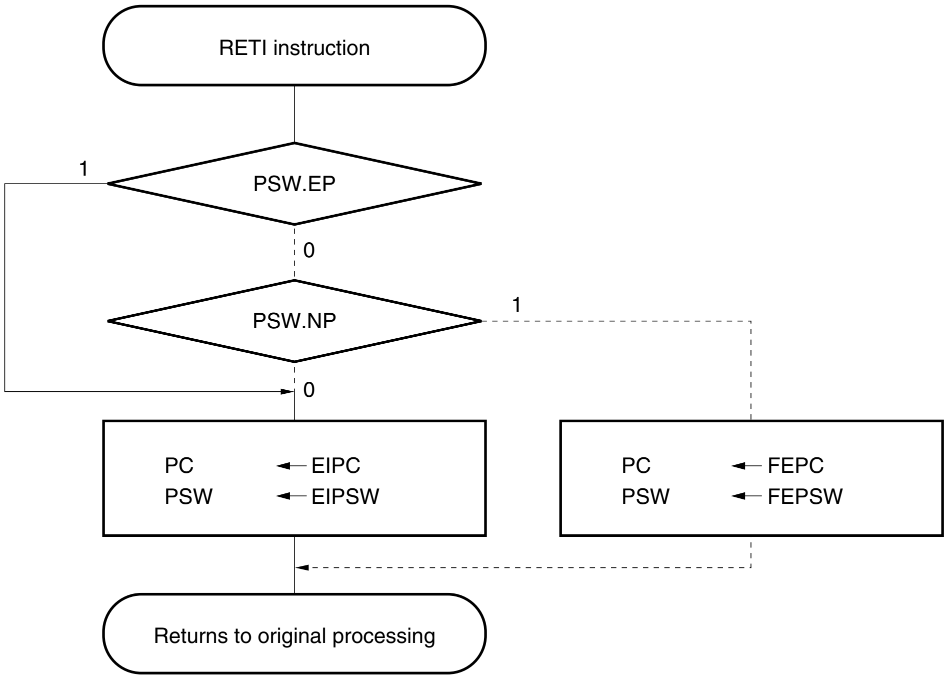
Execution is returned from non-maskable interrupt servicing by using the RETI instruction.

When the RETI instruction is executed, the CPU performs the following processing and transfers control to the address of the return PC.

<1> Loads the saved PC and PSW from EIPC and EIPSW, respectively, because the PSW.EP bit is 1.
<2> Transfers control back to the address of the return PC and PSW.

The processing of the RETI instruction is shown below.

Figure 21-9. RETI Instruction Processing



Caution When the PSW.EP and PSW.NP bits are changed by the LDSR instruction during software exception processing, to restore the PC and PSW correctly when returning by using the RETI instruction, the EP bit must be set (= 1) and the NP bit must be cleared (= 0) using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

21.5.3 Exception status flag (EP)

The EP flag is a status flag that indicates that exception processing is in progress. This flag is set when an exception occurs. The EP flag is assigned to the PSW.

This flag is set to 00000020H after reset.

After reset: 00000020H

	31	8	7	6	5	4	3	2	1	0
PSW	0		NP	EP	ID	SAT	CY	OV	S	Z

EP	Exception processing status
0	Exception processing not in progress
1	Exception processing in progress

21.6 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850E/IG4-H and V850E/IH4-H, an illegal opcode trap (ILGOP: Illegal Opcode Trap) is used as an exception trap.

21.6.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap occurs when an instruction applicable to this illegal instruction is executed.

15 11	10 5	4 0	31 27	26 23	22	16
× × × × ×	1 1 1 1 1 1	× × × × ×	× × × × ×	0 1 1 1 to 1 1 1 1	× × × × × ×	0

×: Arbitrary

Caution Illegal opcodes must not be used because instructions may be newly assigned to these opcodes in the future.

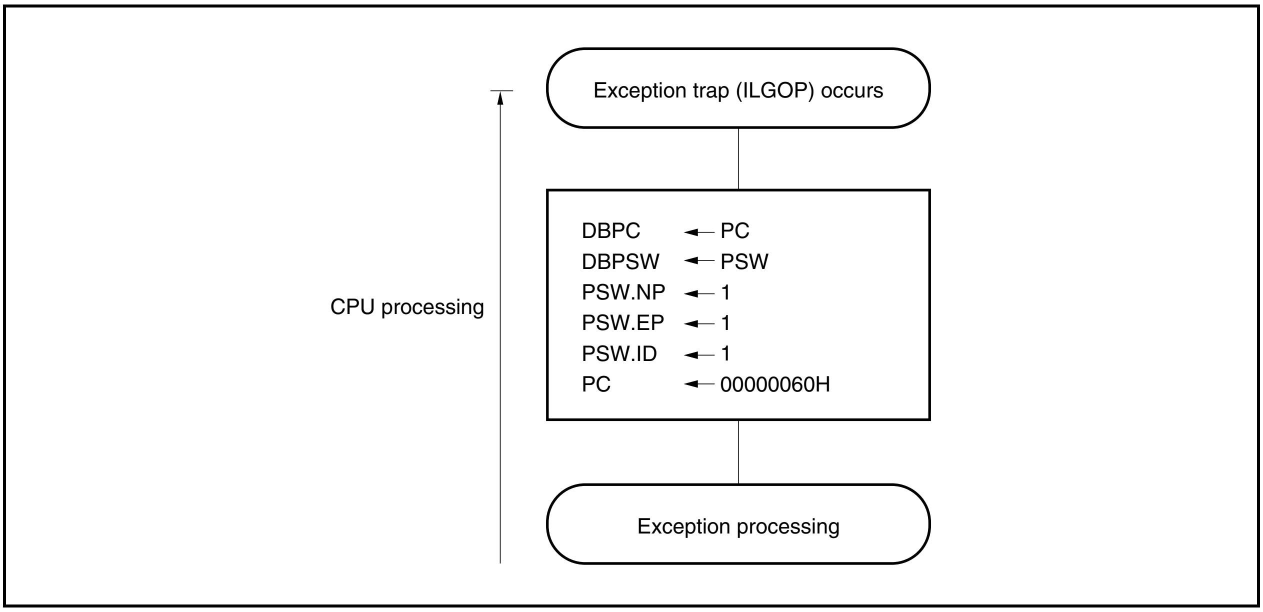
(1) Operation

If an exception trap occurs, the CPU performs the following processing and transfers control to the handler routine.

<1> Saves the current PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP, and PSW.ID bits (1).
<4> Sets the handler address (00000060H) for the exception trap to the PC and transfers control.

The processing of an exception trap is shown below.

Figure 21-10. Exception Trap Processing



(2) Return processing

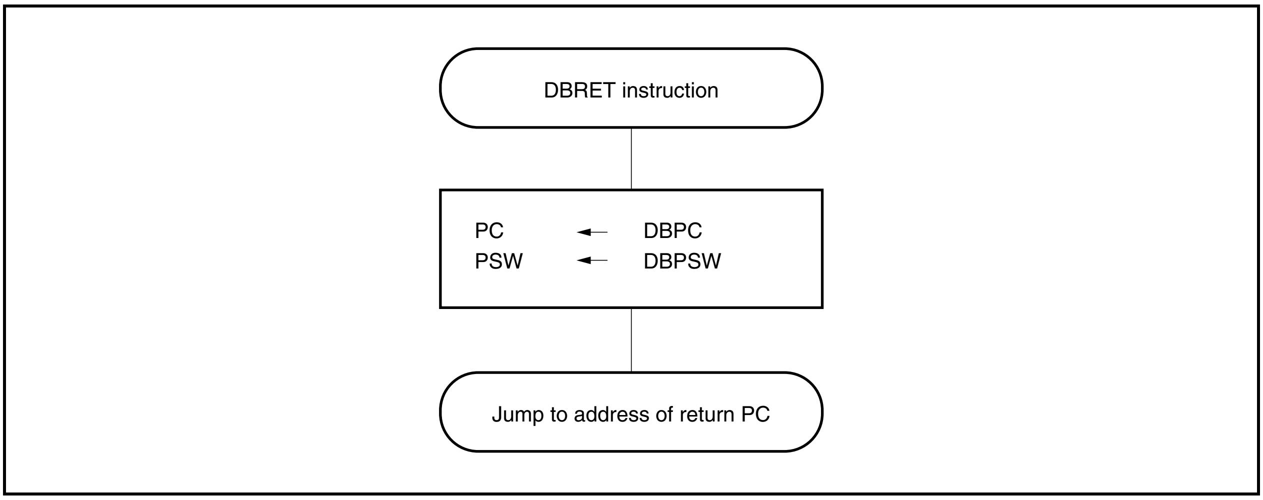
Execution is returned from an exception trap by using the DBRET instruction. When the DBRET instruction is executed, the CPU performs the following processing, and transfers control to the address of the return PC.

<1> Loads the saved PC and PSW from DBPC and DBPSW.
<2> Transfers control back to the address of the return PC and PSW.

Caution DBPC and DBPSW can be accessed only during the interval between the execution of an illegal opcode and the DBRET instruction.

The processing for returning from an exception trap is shown below.

Figure 21-11. Returning from Exception Trap



21.6.2 Debug trap

A debug trap is an exception that can always be acknowledged and occurs when the DBTRAP instruction is executed.

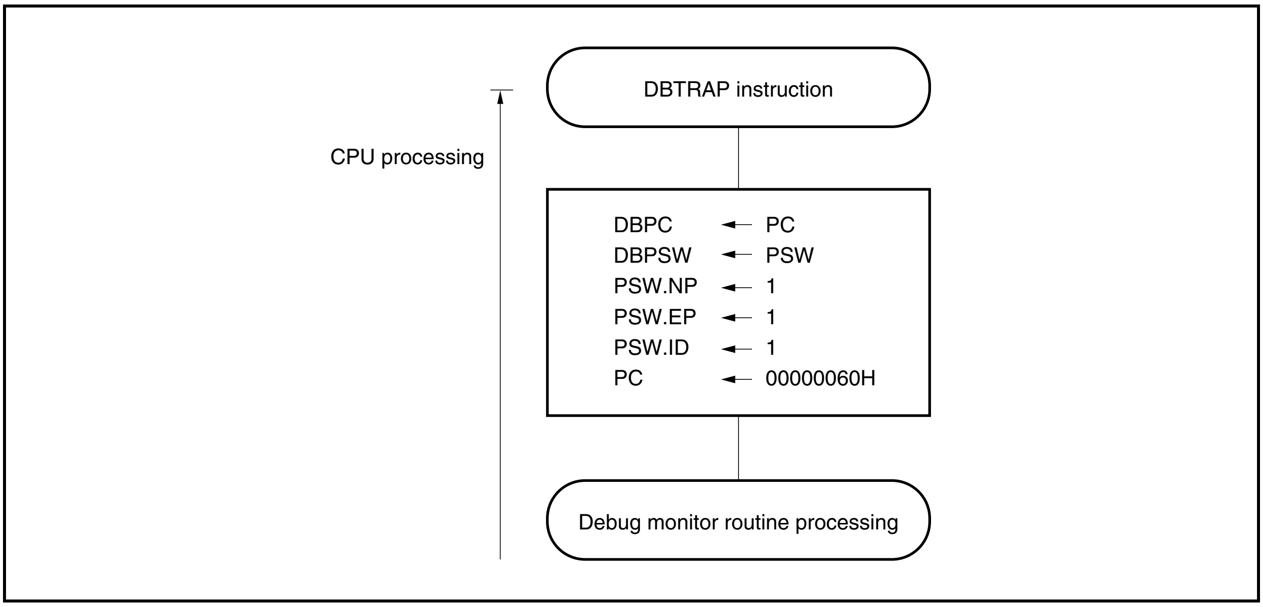
If a debug trap occurs, the CPU performs the following processing.

(1) Operation

<1> Saves the current PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP and PSW.ID bits (1).
<4> Sets the handler address (00000060H) for the debug trap to the PC and transfers control.

The processing of a debug trap is shown below.

Figure 21-12. Debug Trap Processing



(2) Return processing

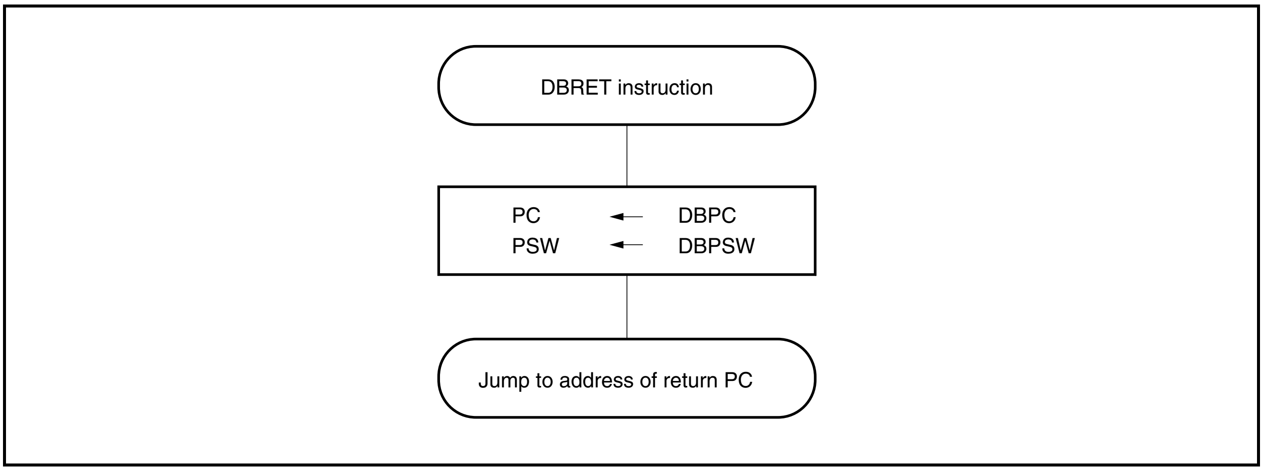
Execution is returned from a debug trap by using the DBRET instruction. When the DBRET instruction is executed, the CPU performs the following processing and transfers control to the address of the return PC.

<1> Loads the return PC and PSW from DBPC and DBPSW.
<2> Transfers control back to the address of the return PC and PSW.

Caution DBPC and DBPSW can be accessed only during the interval between the execution of a DBTRAP instruction and the DBRET instruction.

The processing for returning from a debug trap is shown below.

Figure 21-13. Returning from Debug Trap



21.7 Multiple Interrupt Servicing Control

In multiple interrupt servicing control, the servicing of an interrupt is stopped if an interrupt request signal that has a higher priority level is generated. The higher priority interrupt request signal is then acknowledged and the interrupt is serviced.

If an interrupt request signal with a lower or equal priority level is generated while an interrupt is being serviced, the newly generated interrupt request signal will be held pending.

Multiple interrupt servicing control is performed when interrupts are enabled (PSW.ID bit = 0). Even in an interrupt service routine, multiple interrupt control must be performed while interrupts are enabled (ID bit = 0). If a maskable interrupt or software exception occurs in a maskable interrupt or software exception service program, EIPC and EIPSW must be saved.

The following example shows the procedure for servicing multiple interrupts.

(1) To acknowledge maskable interrupt request signals in a service program

Service program for maskable interrupt or exception

```
          ...
          ...
•  EIPC saved to memory or register
•  EIPSW saved to memory or register
•  EI instruction (interrupt acknowledgment enabled)
          ...
          ...                                          ← Maskable interrupt acknowledgment
          ...
          ...
•  DI instruction (interrupt acknowledgment disabled)
•  Saved value restored to EIPSW
•  Saved value restored to EIPC
•  RETI instruction
```

(2) To generate an exception in a service program

Service program of maskable interrupt or exception

```
          ...
          ...
  • EIPC saved to memory or register
  • EIPSW saved to memory or register
          ...
  • TRAP instruction                    ← Exception such as TRAP instruction acknowledged.
          ...
  • Saved value restored to EIPSW
  • Saved value restored to EIPC
  • RETI instruction
```

Multiple interrupt servicing can be controlled by specifying 8 priority levels for each maskable interrupt request signal (0 to 7: 0 is the highest priority). These levels can be set as desired by using software. The priority order is set by using the xxPRn0 to xxPRn2 bits of the interrupt control request register (xxICn) provided for each maskable interrupt request signal. After a system reset, each interrupt request signal is masked by the corresponding xxMKn bit and its priority order is set to level 7 by the xxPRn0 to xxPRn2 bits.

The priority order of maskable interrupts is as follows.

(High) Level 0 > Level 1 > Level 2 > Level 3 > Level 4 > Level 5 > Level 6 > Level 7 (Low)

Interrupt servicing that has been suspended as a result of multiple servicing control is resumed after the servicing of the higher priority interrupt has been completed and the RETI instruction has been executed.
A pending interrupt request signal is acknowledged after the current interrupt servicing has been completed and the RETI instruction has been executed.

Caution In a non-maskable interrupt service routine (time until the RETI instruction is executed), maskable interrupts are suspended and not acknowledged.

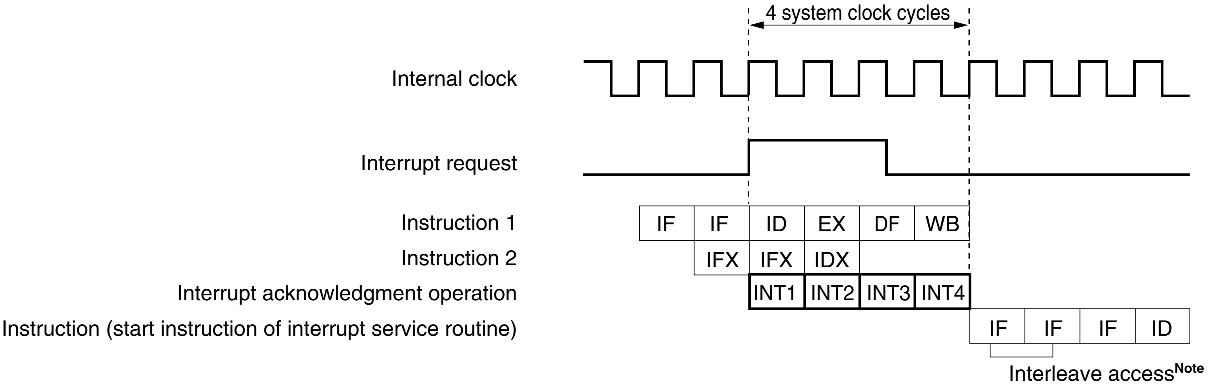
Remark xx: Identification name of each peripheral unit (see **Table 21-2**)
n: Peripheral unit number (see **Table 21-2**)

21.8 Interrupt Response Time of CPU

Except for the following cases, the interrupt response time of the CPU is at least 4 clock cycles. To input interrupt request signals successively, input the next interrupt request signal at least 4 clock cycles after the preceding interrupt.

- In IDLE/STOP mode
- When interrupt request non-sample instructions are successively executed (see **21.9 Periods in Which CPU Does Not Acknowledge Interrupts**.)
- When an on-chip peripheral I/O register is accessed

Figure 21-14. Pipeline Operation When Interrupt Request Signal Is Acknowledged (Outline)



Note For interleave accesses, refer to **8.1.2 2-clock branch** in **V850E1 Architecture User's Manual (U14559E)**.

Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

	Interrupt response time (internal system clock cycles)			Conditions
	Internal interrupt	External interrupt		
		INTP00 to INTP19, INTADT0, INTADT1	INTP00 to INTP02, INTP17 to INTP19	
Minimum	4	4 + Analog filter time	4 + **Note 2** + Digital noise filter	The following cases are exceptions. • In IDLE/STOP mode • When two or more interrupt request non-sample instructions are executed in succession • When an on-chip peripheral I/O register is accessed
Maximum	7[Note 1]	7 + Analog filter time	7 + **Note 2** + Digital noise filter	

Notes 1. When the LD instruction is executed on internal ROM (during align access)

2. For the number of internal system clocks, see **4.6 (1) Digital noise elimination 0 control register n (INTNFCn)**.

Remark a = 00 to 02, 17 to 19

21.9 Periods in Which CPU Does Not Acknowledge Interrupts

An interrupt is acknowledged by the CPU while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (the interrupt is held pending).

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- Store instruction for the command register (PRCMD).
- Store instructions or bit manipulation instructions excluding tst1 instruction for the following registers.
  - Interrupt-related registers:
    Interrupt control register (xxICn) and interrupt mask registers 0 to 6 (IMR0 to IMR6)
  - Power save control register (PSC)

Remark xx: Identification name of each peripheral unit (see **Table 21-2**)
n: Peripheral unit number (see **Table 21-2**)

21.10 Caution

Note that if a port is set to external interrupt input (INTP00 to INTP19, INTADT0, and INTADT1), the timer/counter-related interrupt, serial interface-related interrupt, and A/D converter-related interrupt, which are alternate functions, do not occur.

CHAPTER 22 STANDBY FUNCTION

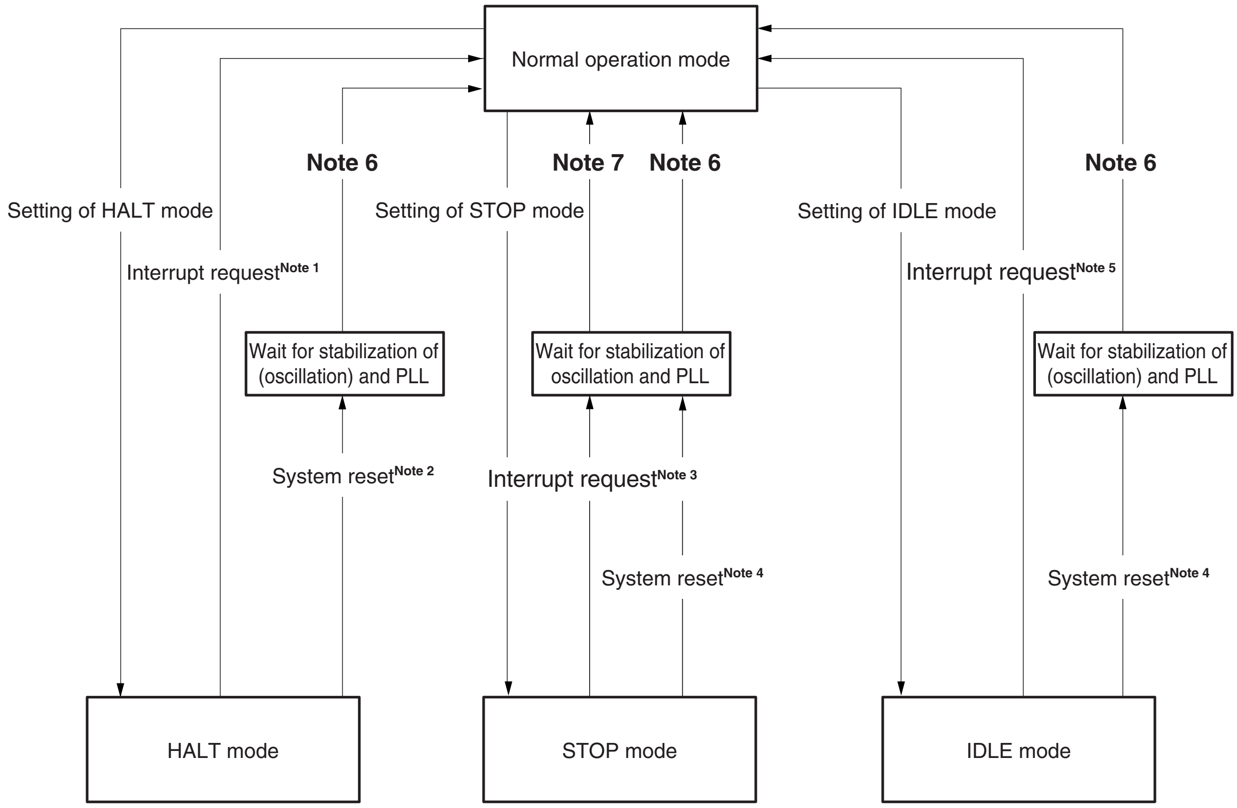
22.1 Overview

The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 22-1.

Table 22-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode to stop only the operating clock of the CPU
IDLE mode	Mode to stop all the operations of the internal circuit except the oscillator, PLL, CSIF in the slave mode, clock monitor, low-voltage detector (LVI), power-on-clear circuit (POC)
STOP mode	Mode to stop all the operations of the internal circuit except the CSIF in the slave mode, low-voltage detector (LVI), power-on-clear circuit (POC)

Figure 22-1. Status Transition


Normal operation mode
Note 6
Note 7
Note 6
Note 6
Setting of HALT mode
Setting of STOP mode
Setting of IDLE mode
Interrupt request[Note 1]
Interrupt request[Note 5]
Wait for stabilization of (oscillation) and PLL
Wait for stabilization of oscillation and PLL
Wait for stabilization of (oscillation) and PLL
System reset[Note 2]
Interrupt request[Note 3]
System reset[Note 4]
System reset[Note 4]
HALT mode
STOP mode
IDLE mode

Notes 1. Non-maskable interrupt request signal (INTWDT) or unmasked maskable interrupt request signal

2. $\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)

3. Unmasked external interrupt request signal (INTP00 to INTP19[Note 8], INTADT0, or INTADT1) or unmasked internal interrupt request signal from (CSIF-related interrupt request signal in the slave mode) peripheral functions operable in STOP mode

4. $\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)

5. Unmasked external interrupt request signal (INTP00 to INTP19[Note 8], INTADT0, or INTADT1) or unmasked internal interrupt request signal (CSIF-related interrupt request signal in the slave mode) from peripheral functions operable in IDLE mode

6. Oscillation stabilization time count by oscillation stabilization time wait control (OST)
The oscillation stabilization time is necessary after release of reset because the PLL is initialized by a reset. The stabilization time is the time determined by default.

7. Oscillation stabilization time count by oscillation stabilization time wait control (OST)
The stabilization time is determined by the setting of the OSTS register.

8. For the INTP00 to INTP02 and INTP17 to INTP19 signals, noise elimination by using an analog filter must be specified.

22.2 Control Registers

(1) Power save control register (PSC)

The PSC register is an 8-bit register that controls the standby function. The STB bit of this register is used to specify the standby mode. This register is a special register (see **3.4.8 Special registers**). This register can be written only by a combination of specific sequences.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF1FEH

	7	6	5	<4>	3	2	<1>	0
PSC	0	0	0	INTM	0	0	STB	0

INTM	Standby mode control[Note 2] by maskable interrupt request (INTxx[Note 1])
0	Standby mode release by INTxx request enabled
1	Standby mode release by INTxx request disabled

STB	Sets operation mode
0	Normal mode
1	Standby mode

Notes 1. For details, see **Table 21-1 Interrupt Source List**.

2. The setting is valid only in the IDLE mode and STOP mode.

Cautions 1. Be sure to set bits 0, 2, 3, and 5 to 7 to “0”.

2. Before setting a standby mode by setting the STB bit to 1, be sure to set the PCC register to 03H and then set the STB bit to 1. Otherwise, the standby mode may not be set or released. After releasing the standby mode, change the value of the PCC register to the desired value.

3. To set the IDLE mode or STOP mode, set the PCC register to 03H, and the PSMR.PSM0 bit in that order and then set the STB bit to 1.

(2) Power save mode register (PSMR)

The PSMR register is an 8-bit register that controls the operation in the software standby mode.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	0	0	0	0	0	0	0	PSM0

PSM0	Operation in software standby mode specification
0	IDLE mode
1	STOP mode

Cautions 1. Be sure to set bits 1 to 7 to "0".

2. The PSM0 bit is valid only when the PSC.STB bit is 1.

22.3 HALT Mode

22.3.1 Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

When HALT mode is set, clock supply is stopped to the CPU only. The clock generator and PLL continue operating. Clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 22-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

Cautions 1. Insert five or more NOP instructions after the HALT instruction.

2. If the HALT instruction is executed while an interrupt request is being held pending, the HALT mode is set but is released immediately by the pending interrupt request.

22.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (INTWDT), an unmasked maskable interrupt request signal, and a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)).

After the HALT mode has been released, the normal operation mode is restored.

(1) Releasing HALT mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The HALT mode is released by a non-maskable interrupt request signal (INTWDT) or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request signal with a priority lower than or same as the interrupt currently being serviced is generated, the HALT mode is released, but the newly generated interrupt request signal is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the HALT instruction.

(b) If an interrupt request signal with a priority higher than that of the interrupt currently being serviced is issued (including a non-maskable interrupt request signal), the HALT mode is released and that interrupt request signal is acknowledged. Therefore, execution branches to the handler address.

Table 22-2. Operation After Releasing HALT Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address	
Unmasked maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing HALT mode by $\overline{\text{RESET}}$ pin input or by WDTRES, LVIRES, or POCRES signal generation

The same operation as the normal reset operation is performed.

Table 22-3. Operation Status in HALT Mode

Item \ Setting of HALT Mode		Operation Status
Clock generator, PLL		Operates
System clock (f_{XX})		Supply
CPU		Stops operation
DMA		Operable
Interrupt controller		Operable
Timer	TAA0 to TAA2	Operable
	TAB0, TAB1	Operable
	TMT0 to TMT3	Operable
	TMM0 to TMM3	Operable
Watchdog timer		Operable
Serial interface	CSIF0 to CSIF2	Operable
	UARTA0 to UARTA2	Operable
	UARTB	Operable
	I^{2}C	Operable
A/D converters 0 to 2		Operable
Clock monitor		Operable
Low-voltage detector		Operable
Power-on-clear circuit		Operable
USB function		Operable
Port function		Retains status before HALT mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the HALT mode was set.

22.4 IDLE Mode

22.4.1 Setting and operation status

The IDLE mode is set by clearing (0) the PSMR.PSM0 bit and setting (1) the PSC.STB bit in the normal operation mode.

In the IDLE mode, the clock generator and PLL continue operation but clock supply to the CPU and other on-chip peripheral functions stops.

As a result, program execution stops and the contents of the internal RAM before the IDLE mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with an external clock continue operating.

Table 22-5 shows the operation status in the IDLE mode.

The IDLE mode can reduce the power consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The clock generator and PLL do not stop, so the normal operation mode can be restored without waiting for the oscillation stabilization time after the IDLE mode has been released, in the same manner as when the HALT mode is released.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE mode.

22.4.2 Releasing IDLE mode

The IDLE mode is released by an unmasked external interrupt request signal (INTP00 to INTP19[Note], INTADT0, or INTADT1 pin input), an unmasked internal interrupt request signal (CSIF-related interrupt request signal in the slave mode) from the peripheral functions operable in the IDLE mode, or a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)).

After the IDLE mode has been released, the normal operation mode is restored.

Note For the INTP00 to INTP02 and INTP17 to INTP19 signals, noise elimination by using an analog filter must be specified.

(1) Releasing IDLE mode by unmasked maskable interrupt request signal

The IDLE mode is released by an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the IDLE mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

Caution When PSC.INTM bit = 1, the IDLE mode cannot be released by the unmasked maskable interrupt request signal.

(a) If an interrupt request with a priority lower than or same as the interrupt request signal currently being serviced is generated, the IDLE mode is released, but the newly generated interrupt is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the IDLE instruction.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the IDLE mode is released and that interrupt request signal is acknowledged. Therefore, execution branches to the handler address.

Table 22-4. Operation After Releasing IDLE Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Unmasked maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing IDLE mode by $\overline{\text{RESET}}$ pin input or by LVIRES or POCRES signal generation

The same operation as the normal reset operation is performed.

Table 22-5. Operation Status in IDLE Mode

Setting of IDLE Mode / Item		Operation Status
Clock generator, PLL		Operates
System clock (f_{XX})		Stops supply
CPU		Stops operation
DMA		Stops operation
Interrupt controller		Stops operation
Timer	TAA0 to TAA2	Stops operation
	TAB0, TAB1	Stops operation
	TMT0 to TMT3	Stops operation
	TMM0 to TMM3	Stops operation
Watchdog timer		Stops operation
Serial interface	CSIF0 to CSIF2	Operable when $\overline{\text{SCKFn}}$ input clock is selected as count clock (in slave mode) (n = 0 to 2)
	UARTA0 to UARTA2	Stops operation
	UARTB	Stops operation
	I^2C	Stops operation
A/D converters 0 to 2		Stops operation
Clock monitor		Operable
Low-voltage detector		Operable
Power-on-clear circuit		Operable
USB function		Stops operation
Port function		Retains status before IDLE mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE mode was set.

22.5 STOP Mode

22.5.1 Setting and operation status

The STOP mode is set by setting (1) the PSMR.PSM0 bit and setting (1) the PSC.STB bit in the normal operation mode.

In the STOP mode, the clock generator stops operation. Clock supply to the CPU and the on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the STOP mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with an external clock continue operating.

Table 22-7 shows the operation status in the STOP mode.

Because the STOP stops operation of the clock generator, it reduces the power consumption to a level lower than the IDLE mode. When the external clock is not used, the power consumption can be minimized with only leakage current flowing.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.

22.5.2 Releasing STOP mode

The STOP mode is released by an unmasked external interrupt request signal (INTP00 to INTP19[Note], INTADT0, or INTADT1 pin input), an unmasked internal interrupt request signal (CSIF-related interrupt signal in the slave mode) from the peripheral functions operable in the STOP mode, or a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)).

After the STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

Note For the INTP00 to INTP02 and INTP17 to INTP19 signals, noise elimination by using an analog filter must be specified.

(1) Releasing STOP mode by unmasked maskable interrupt request signal

The STOP mode is released by an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

Caution When PSC.INTM bit = 1, the STOP mode cannot be released by the unmasked maskable interrupt request signal.

(a) If an interrupt request with a priority lower than or same as the interrupt request currently being serviced is generated, the STOP mode is released, but the newly generated interrupt is not acknowledged. The interrupt request itself is retained. Therefore, execution starts at the next instruction after the STOP instruction.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued, the STOP mode is released and that interrupt request is acknowledged. Therefore, execution branches to the handler address.

Table 22-6. Operation After Releasing STOP Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Unmasked maskable interrupt request	Execution branches to the handler address or the next instruction is executed after securing oscillation stabilization time	The next instruction is executed after securing oscillation stabilization time

(2) Releasing STOP mode by $\overline{\text{RESET}}$ pin input or by LVIRES or POCRES signal generation

The same operation as the normal reset operation is performed.

Table 22-7. Operation Status in STOP Mode

Setting of STOP Mode Item		Operation Status
Clock generator, PLL		Stops operation
System clock (f_{XX})		Stops supply
CPU		Stops operation
DMA		Stops operation
Interrupt controller		Stops operation
Timer	TAA0 to TAA2	Stops operation
	TAB0, TAB1	Stops operation
	TMT0 to TMT3	Stops operation
	TMM0 to TMM3	Stops operation
Watchdog timer		Stops operation
Serial interface	CSIF0 to CSIF2	Operable when $\overline{\text{SCKFn}}$ input clock is selected as count clock (in slave mode) (n = 0 to 2)
	UARTA0 to UARTA2	Stops operation
	UARTB	Stops operation
	I^{2}C	Stops operation
A/D converters 0 to 2		Stops operation
Clock monitor		Stops operation
Low-voltage detector		Operable
Power-on-clear circuit		Operable
USB function		Stops operation
Port function		Retains status before STOP mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the STOP mode was set.

22.6 Securing Oscillation Stabilization Time

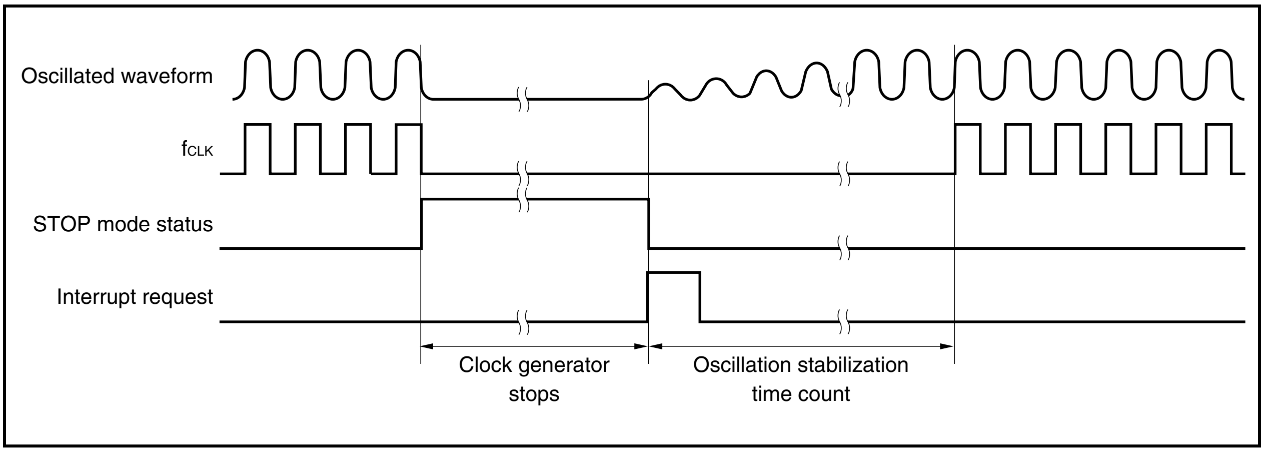
When the STOP mode is released, the oscillation stabilization time set by the OSTS register elapses. The oscillation stabilization time is the reset value of the OSTS register, $2^{15}/f_X$ (2.62 ms at f_X = 12.5 MHz), if the STOP mode is released by $\overline{\text{RESET}}$ pin input.

However, the actual oscillation stabilization time for the resonator is 1.311 ms (when f_X = 12.5 MHz), and the other half of the time is consumed in stabilization of the PLL. When exiting the STOP mode, therefore, specify an oscillation stabilization time double that required for the used resonator to stabilize. In addition, when releasing the mode by $\overline{\text{RESET}}$ pin input, be sure to secure the oscillation stabilization time by outputting the $\overline{\text{RESET}}$ signal at low level for the time longer than the oscillation stabilization time of the used resonator minus the fixed oscillation stabilization time.

The timer for counting the oscillation stabilization time secures oscillation stabilization time equal to the overflow time of the watchdog timer.

The operation performed when the STOP mode is released by an interrupt request signal is shown below.

Figure 22-2. Oscillation Stabilization Time



Caution For details of the OSTS register, see 5.3 (5) Oscillation stabilization time select register (OSTS).

CHAPTER 23 RESET FUNCTIONS

23.1 Overview

- System reset by $\overline{\text{RESET}}$ pin input
- System reset signal (WDTRES) generation by watchdog timer (WDT) overflow
- System reset signal (LVIRES) generation by low-voltage detector (LVI)
- System reset signal (POCRES) generation by power-on-clear circuit (POC)
- Forced reset by on-chip debug function (DCU) and reset mask function (see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.)

23.2 Control Register

(1) Reset source flag register (RESF)

The RESF register is an 8-bit register that indicates occurrence of a reset request from the watchdog timer (WDT) or low-voltage detector (LVI).

The WDTRF or LVIRF bit of this register is set to 1 when the internal reset source signal from WDT or LVI is asserted. The WDTRF or LVIRF bit is cleared by a reset signal (the one generated by inputting the $\overline{\text{RESET}}$ pin, the POCRES signal generated by the power-on-clear circuit (POC), or the forced reset signal generated by the on-chip debug function), a bit manipulation instruction, or a store instruction (writing 0 to the WDTRF or LVIRF bit).

The RESF register is a special register and can be written only in a combination of specific sequences (see **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units. However, bits 0 and 4 can only be cleared (0) by writing.

This register is set to 00H by $\overline{\text{RESET}}$ pin input and reset by the power-on-clear circuit (POC). This register is set to 00H by RESET pin input, a reset by the power-on-clear circuit (POC), or a forced reset by the on-chip debug function. For details on reset conflict, see **Cautions** on the next page.

After reset: 00H[Note] R/W Address: FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LIVRF

WDTRF	Occurrence of reset signal from watchdog timer (WDT)
0	Read: No reset request, Write: Clear
1	Reset request

LIVRF	Occurrence of reset signal from low-voltage detector (LVI)
0	Read: No reset request, Write: Clear
1	Reset request

Note After a reset by $\overline{\text{RESET}}$ pin input or the power-on-clear circuit (POC), or after a forced reset by the on-chip debug function: 00H
After a reset due to a watchdog timer overflow: 10H
After a reset by the low-voltage detector (LVI): 01H

Cautions 1. If setting (occurrence of reset of set source) and clearing (occurrence of system reset or writing 0 to the WDTRF or LVIRF bit) of the RESF register conflict, the priorities are as follows.

<1> A reset by $\overline{\text{RESET}}$ pin input or the power-on-clear circuit (POC), or a forced reset by the on-chip debug function (that clears the RESF register)
<2> A reset by WDT or LVI (that sets the RESF register)
<3> Writing 0 to the WDTRF or LVIRF bit by a bit manipulation or store instruction (that clears the RESF register)

2. Even if reset masking was specified when a flag setting source occurred, the flag is set. (Reset masking does not affect setting the flag.)

23.3 Operation

(1) Reset operation by $\overline{\text{RESET}}$ pin input

When a low level is input to the $\overline{\text{RESET}}$ pin, the V850E/IG4-H and V850E/IH4-H are reset, and each hardware unit is initialized to a specific status.

The oscillator continues oscillation even while a low level is input to the $\overline{\text{RESET}}$ pin but the oscillation mode is initialized to the clock-through mode (PLLCTL register = 01H) and the CPU clock (f_{CPU}) division to $f_{XX}/8$ (PCC register = 03H).

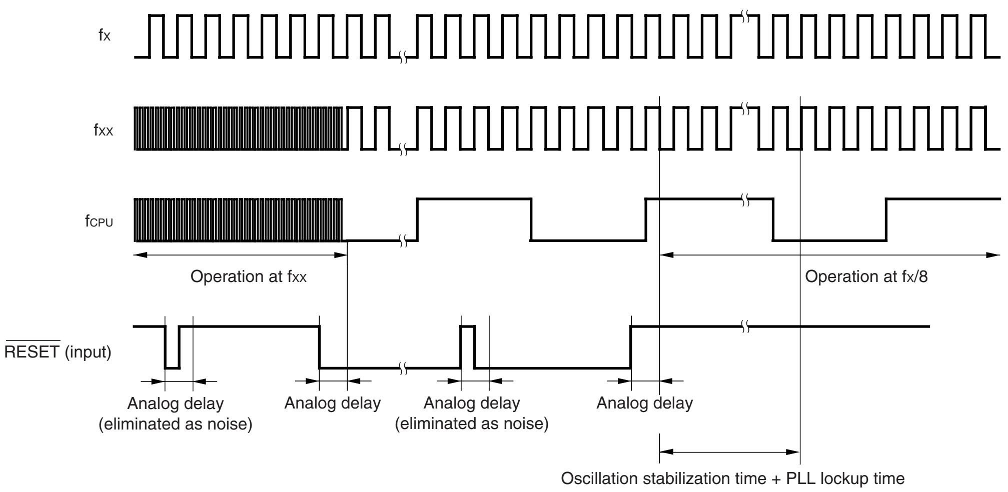
The reset status is released when the $\overline{\text{RESET}}$ pin input goes from low to high. After the reset status is released, the oscillation stabilization time of the oscillator and lockup time of PLL (default value of OSTS register for the total time: $2^{15}/f_X$ (2.62 ms (f_X = 12.5 MHz)) elapse, and then the CPU starts program execution. After release of reset, therefore, the operation is started in the clock-through mode and at $f_{XX}/8$.

The status of each hardware unit during the reset period and after the reset status is released is shown below.

Hardware	During Reset Period	After Reset Is Released
Clock generator: Oscillator (f_X) Internal system clock (f_{CLK}) CPU clock (f_{CPU})	Oscillation/supply continues However, the CPU clock (f_{CPU}) is initialized to $f_{XX}/8$.	
Clock generator: Peripheral clock (f_{XX} to $f_{XX}/4096$)	Oscillation/supply stops	Oscillation/supply starts after securing of oscillation stabilization time
Clock generator: Watchdog timer clock ($f_{XX}/1024$)	Oscillation/supply stops	Oscillation/supply starts
CPU	Initialized	Program execution starts after securing of oscillation stabilization time
Internal RAM	Retains value immediately before reset input only in the STOP mode during reset input. Otherwise, undefined.	
Ports (including alternate-function pins)	High impedance	
On-chip peripheral I/O registers (other than ports)	Initialized to specific status	
On-chip peripheral functions other than above	Stops operation	Can start operation

The reset operation by $\overline{\text{RESET}}$ pin input is illustrated below.

Figure 23-1. Reset Operation by $\overline{\text{RESET}}$ Pin Input



Caution After release of reset, make sure that the oscillation stabilization time (1.311 ms (at fx = 12.5 MHz)) and PLL lockup time (1.311 ms (at fx = 12.5 MHz)) elapse. If an oscillation stabilization time of 1.311 ms is not sufficient to secure stable oscillation, keep the $\overline{\text{RESET}}$ pin low for the deficient time.

Remark The relationship between fxx and fx in the above timing chart is fxx = 8 × fx.

The operation after release of reset is the same in both the PLL mode and clock-through mode and is started in the clock-through mode. Set the PLL mode by software control (setting PLLCTL.SELPLL bit to 1). To improve noise immunity, it is recommended to set the PLL mode and then speed up the CPU clock (example: PCC register = 00H (fxx operation)).

(2) Reset operation (WDTRES) by overflow of watchdog timer (WDT)

If the reset mode is set to reset upon overflow of the watchdog timer (WDT) (WDTM.WDM1 and WDTM.WDM0 bits = 10 or 11), the system is reset and each hardware is initialized to a specific state when WDT overflows (WDTRES).

If the WDTRES signal is generated, the RESF.WDTRF bit is set to 1, indicating that internal reset has occurred.

The operations during the reset period and after release of reset, other than the operation of the RESF register, are the same as the reset operation by $\overline{\text{RESET}}$ pin input (see **(1) Reset operation by $\overline{\text{RESET}}$ pin input**).

(3) Reset operation (LVIRES) by low-voltage detector (LVI)

When LVI operation is enabled, the supply voltage (V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}, V850E/IH4-H: FV_{DD}) and detection voltage (V_{LVI}) are compared and if the supply voltage drops below the detection voltage, the system is reset (when the LVIM.LVIMD bit is set to “1”) and each hardware is initialized to a specific state.

The system is reset when the supply voltage drops below the detection voltage and the reset ends when the supply voltage is equal to or exceeds the detection voltage. After the reset ends, when the oscillation stabilization time (default value of the OSTS register: $2^{15}/f_X$) of the oscillator has elapsed, the CPU starts executing the program.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the $\overline{\text{RESET}}$ pin (see **(1) Reset operation by $\overline{\text{RESET}}$ pin input**).

For details of the reset operation by low-voltage detector (LVI), see **CHAPTER 24 LOW-VOLTAGE DETECTOR**.

(4) Reset operation (POCRES) by power-on-clear circuit (POC)

When the supply voltage (V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}, V850E/IH4-H: FV_{DD}) and detection voltage (V_{POC0}) are compared and if the supply voltage drops below the detection voltage (including at power application), the system is reset and each hardware is initialized to a specific state.

The system is reset when the supply voltage drops below the detection voltage and the reset ends when the supply voltage is equal to or exceeds the detection voltage. After the reset ends, when the oscillation stabilization time (default value of the OSTS register: $2^{15}/f_X$) of the oscillator has elapsed, the CPU starts executing the program.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the $\overline{\text{RESET}}$ pin (see **(1) Reset operation by $\overline{\text{RESET}}$ pin input**).

For details of the reset operation by power-on-clear circuit (POC), see **CHAPTER 25 POWER-ON-CLEAR CIRCUIT**.

CHAPTER 24 LOW-VOLTAGE DETECTOR

24.1 Functions

The low-voltage detector (LVI) has the following functions.

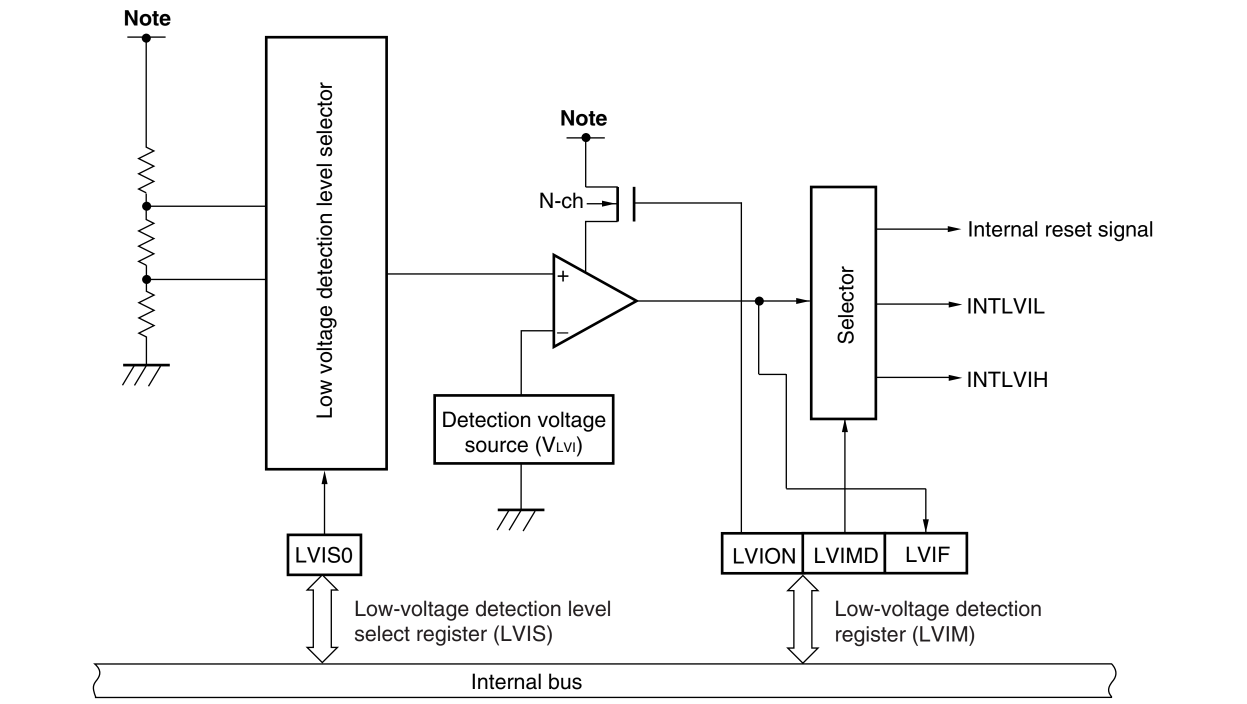
- Compares the supply voltage (V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}, V850E/IH4-H: FV_{DD}) and detection voltage (V_{LVI}) and generates an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) when the supply voltage drops below the detection voltage.
- The level of the supply voltage to be detected can be changed by software (in two steps).
- An interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) can be selected.
- Can operate in STOP mode.
- Operation can be stopped by software.

If the low-voltage detector is used to generate a reset signal, the RESF.LVIRF bit is set to 1 when the reset signal is generated. For details of RESF register, see **CHAPTER 23 RESET FUNCTIONS**.

24.2 Configuration

The block diagram is shown below.

Figure 24-1. Block Diagram of Low-Voltage Detector


Note
Low voltage detection level selector
Note
N-ch
+
–
Detection voltage source (VLVI)
Selector
Internal reset signal
INTLVIL
INTLVIH
LVIS0
LVION
LVIMD
LVIF
Low-voltage detection level select register (LVIS)
Low-voltage detection register (LVIM)
Internal bus

Note V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}
V850E/IH4-H: FV_{DD}

24.3 Control Registers

(1) Low-voltage detection register (LVIM)

The LVIM register is used to enable or disable low voltage detection, and to set the operation mode of the low-voltage detector. The LVIM register is a special register. It can be written only by a combination of specific sequences (see **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

Reset other than reset by the low-voltage detector (LVI) sets this register to 00H.

After reset: 00H R/W Address: FFFFF890H

	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	Low voltage detection operation enable or disable
0	Disable operation.
1	Enable operation.

LVIMD	Selection of operation mode of low voltage detection
0	Generate interrupt request signal INTLVIL when supply voltage < detection voltage. Generate interrupt request signal INTLVIH when supply voltage > detection voltage.
1	Generate internal reset signal LVIRES when supply voltage < detection voltage.

LVIF	Low voltage detection flag
0	When supply voltage > detection voltage, or when operation is disabled
1	Supply voltage < detection voltage

Cautions 1. After setting the LVION bit to 1, wait for 0.1 ms or more before checking the voltage using the LVIF bit.

2. The value of the LVIF flag is output as the output signals INTLVIL or INTLVIH when the LVION bit = 1 and LVIMD bit = 0.

3. If the LVION bit = 1 and LVIMD bit = 1, the low-voltage detector (LVI) cannot be stopped until a reset request other than that of by the LVI is generated.

4. Be sure to set bits 2 to 6 to “0”.

(2) Low-voltage detection level select register (LVIS)

The LVIS register is used to select the level of low voltage to be detected.

This register can be read or written in 8-bit units.

Reset other than reset by the low-voltage detector (LVI) sets this register to 00H.

After reset: 00H R/W Address: FFFFF891H

	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	Detection level
0	4.4 V ±0.2 V
1	4.2 V ±0.2 V

Cautions 1. The LVIS register cannot be written until a reset request due to something other than the low-voltage detector (LVI) is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1.

2. Be sure to clear bits 1 to 7 to “0”.

24.4 Operation

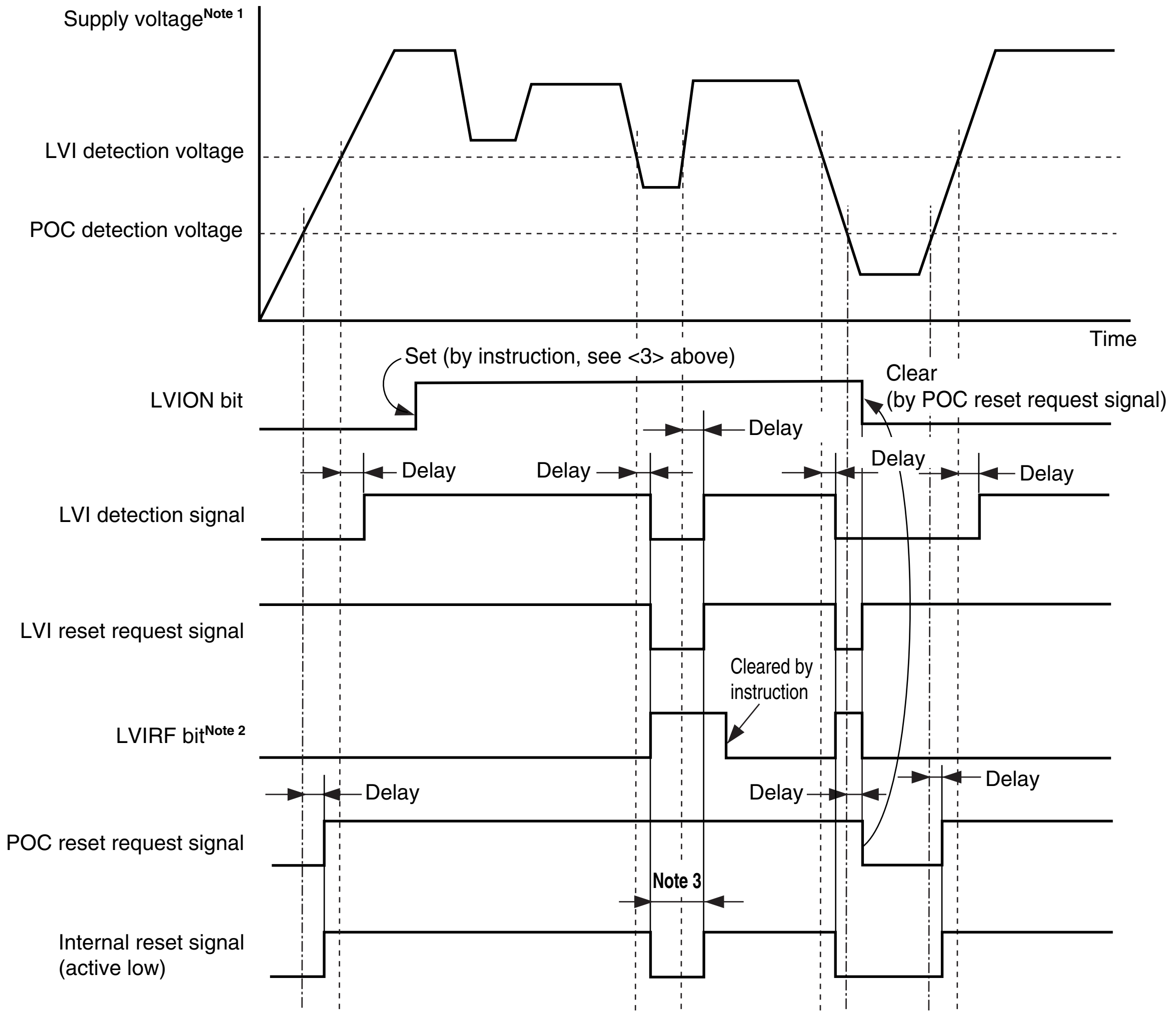
Depending on the setting of the LVIM.LVIMD bit, an interrupt request signal (INTLVIL, INTLVIH) or an internal reset signal (LVIRES) is generated.

24.4.1 To use for internal reset signal

<To start operation>
<1> Mask the interrupt of the low-voltage detector (LVI).
<2> Select the voltage to be detected by using the LVIS.LVIS0 bit.
<3> Set the LVIM. LVION bit to 1 (to enable operation).
<4> Insert a wait cycle of 0.1 ms or more by software.
<5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.
<6> Set the LVIM.LVIMD bit to 1 (to generate an internal reset signal).

Caution If the LVIMD bit is set to 1, the contents of the LVIM and LVIS registers cannot be changed until a reset request other than the low-voltage detector (LVI) is generated.

Figure 24-2. Operation Timing of Low-Voltage Detector (LVIMD Bit = 1)



Notes 1. V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}
V850E/IH4-H: FV_{DD}

2. The LVIRF bit is bit 0 of the reset source flag register (RESF). For details of RESF, see **CHAPTER 23 RESET FUNCTIONS**.

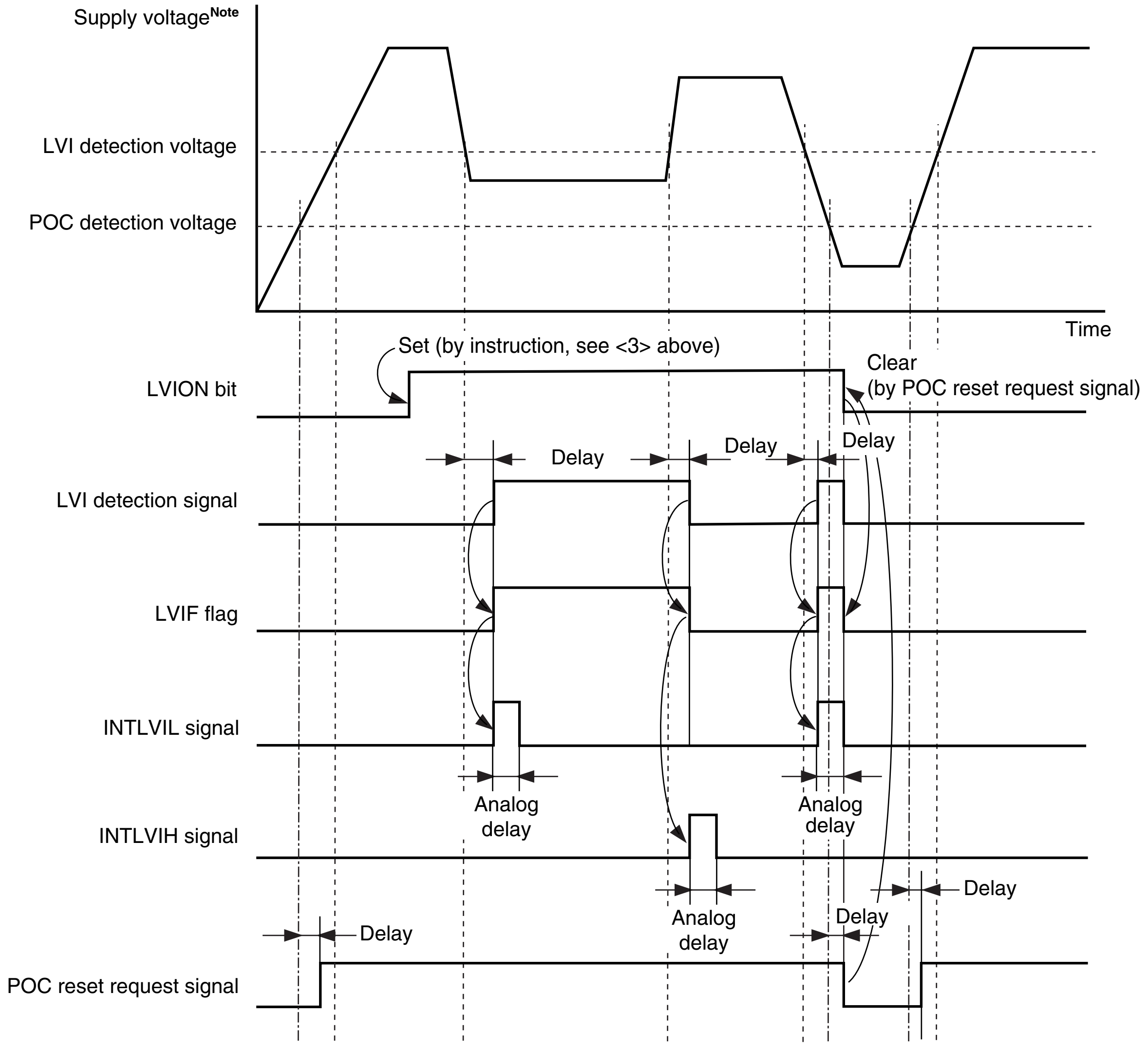
3. During the period in which the supply voltage is the set voltage or lower, the internal reset signal is retained (internal reset state).

24.4.2 To use for interrupt

<To start operation>
<1> Mask the interrupt of the low-voltage detector (LVI).
<2> Select the voltage to be detected by using the LVIS.LVIS0 bit.
<3> Set the LVIM.LVION bit to 1 (to enable operation).
<4> Insert a wait cycle of 0.1 ms or more by software.
<5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.
<6> Clear the interrupt request flag of LVI.
<7> Unmask the interrupt of LVI.

<To stop operation>
Set the LVION bit to 0.

Figure 24-3. Operation Timing of Low-Voltage Detector (LVIMD Bit = 0)



Note V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}
V850E/IH4-H: FV_{DD}

CHAPTER 25 POWER-ON CLEAR CIRCUIT

25.1 Function

Functions of the power-on-clear circuit (POC) are shown below.

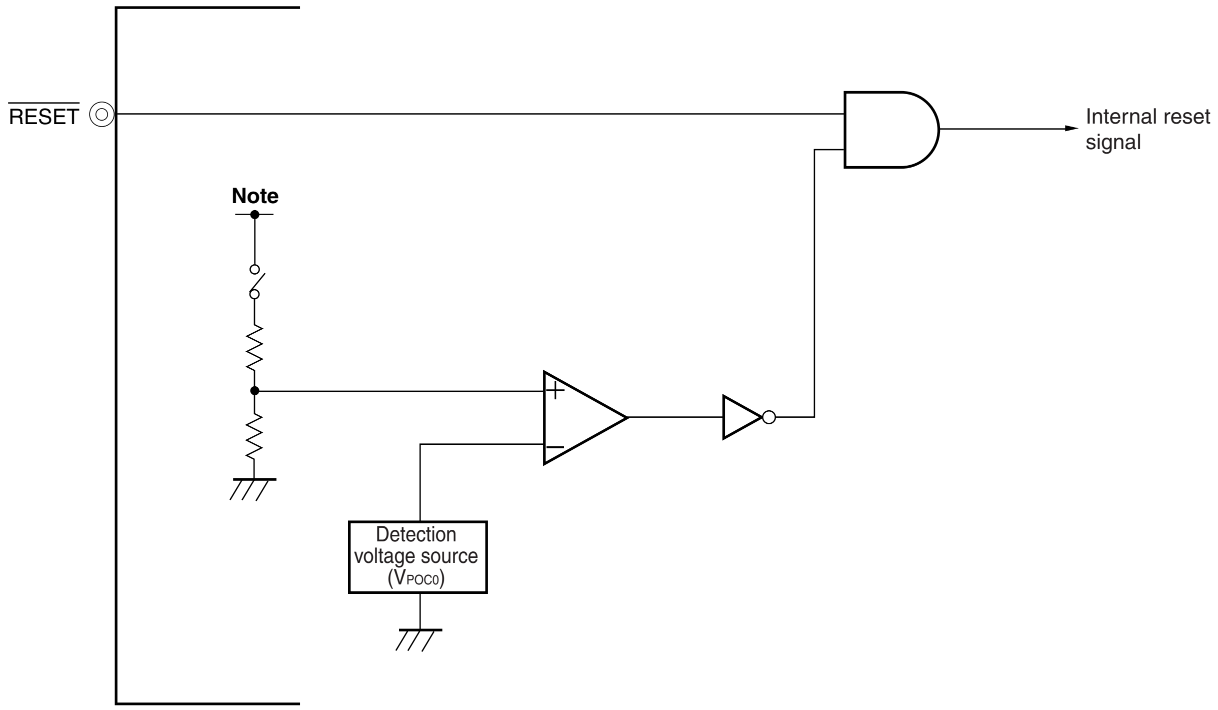
- Generates a reset signal (POCRES) upon power application.
- Compares the supply voltage (V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}, V850E/IH4-H: FV_{DD}) and detection voltage (V_{POC0}), and generates a reset signal when the supply voltage drops below the detection voltage (detection voltage (V_{POC0}): 3.7 V ±0.2 V).

Remark The V850E/IG4-H and V850E/IH4-H have the reset source flag register (RESF) that indicates generation of a reset signal (WDTRES) by watchdog timer overflow and a reset signal (LVIRES) by low-voltage detector (LVI).
The RESF register is not cleared to 00H when a reset signal (WDTRES or LVIRES) is generated, and its flag corresponding to the reset source is set to 1.
The RESF register is cleared (00H) when a reset signal (POCRES) by power-on-clear circuit (POC) is generated.
For details of the RESF register, see **CHAPTER 23 RESET FUNCTIONS**.

25.2 Configuration

The block diagram is shown below.

Figure 25-1. Block Diagram of Power-on-Clear Circuit


$\overline{\text{RESET}}$
Note
Internal reset signal
+
−
Detection voltage source (V_{POC0})

Note V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}
V850E/IH4-H: FV_{DD}

25.3 Operation

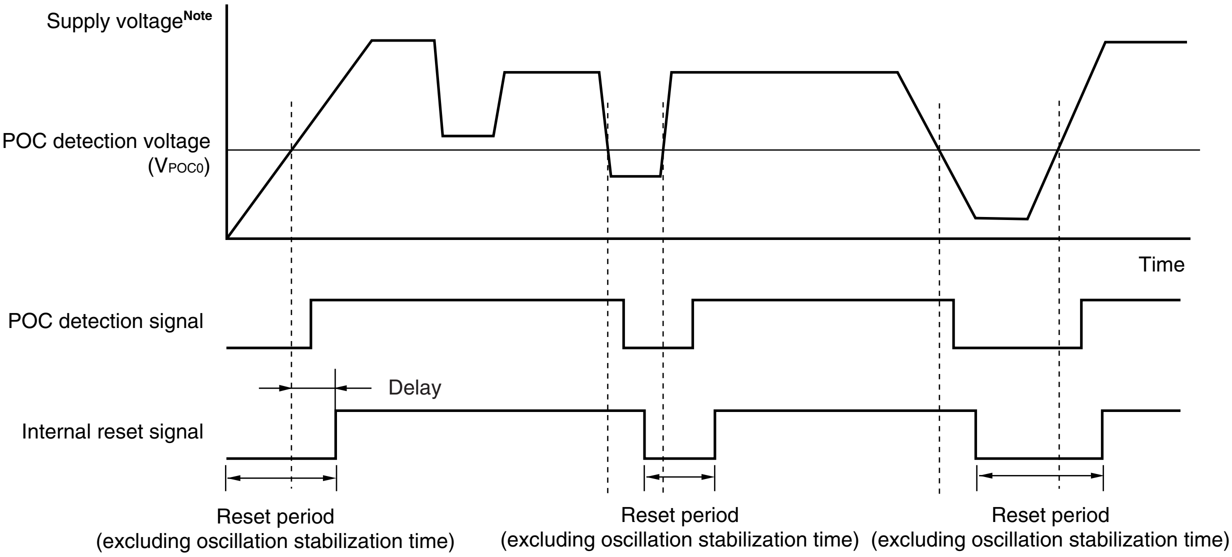
When the supply voltage and detection voltage are compared and if the supply voltage drops below the detection voltage (including at power application), the system is reset and each hardware is initialized to the specific status.

The system is reset from when low voltage is detected until the supply voltage becomes higher than the detection voltage. After a reset is released, when the oscillation stabilization time (default value of the OSTS register: $2^{15}/fx$) of the oscillator has elapsed, the CPU starts executing the program.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the $\overline{\text{RESET}}$ pin (see **23.3 (1) Reset operation by $\overline{\text{RESET}}$ pin input**).

The following shows the timing chart.

Figure 25-2. Timing of Reset Signal Generation by Power-on-Clear Circuit



Note V850E/IG4-H: EV_{DD0}, EV_{DD1}, EV_{DD2}
V850E/IH4-H: FV_{DD}

CHAPTER 26 ON-CHIP DEBUG FUNCTION

The on-chip debug function of the V850E/IG4-H and V850E/IH4-H can be realized in the following three ways.

<1> Debugging using the DCU (debug control unit) (with trace function) (V850E/IH4-H only): using on-chip debug emulator product of partner
By using the $\overline{\text{DRST}}$, DCK, DMS, DDI, DDO, TRCCLK, TRCDATA0 to TRCDATA3, and TRCEND pins as debug interface pins, on-chip debugging is realized by the internal DCU of the V850E/IH4-H.

<2> Debugging using the DCU (no trace function): using MINICUBE
By using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins as debug interface pins, on-chip debugging is realized by the internal DCU.

<3> Debugging without using the DCU: using MINICUBE2
On-chip debugging is realized by MINICUBE2 without using the DCU but by using the user resources.

The following table shows the features of the three on-chip debug functions.

Table 26-1. On-Chip Debug Function Features

		Debugging Using DCU (<1>)	Debugging Using DCU (<2>)	Debugging Without Using DCU (<3>)
Target product		V850E/IH4-H	V850E/IG4-H, V850E/IH4-H	V850E/IG4-H, V850E/IH4-H
Debug interface pins		$\overline{\text{DRST}}$, DCK, DMS, DDI, DDO, TRCCLK, TRCDATA0 to TRCDATA3, TRCEND	$\overline{\text{DRST}}$, DCK, DMS, DDI, DDO	• When UARTA0 is used RXDA0, TXDA0 • When CSIF0 is used SIF0, SOF0, $\overline{\text{SCKF0}}$, HS (P44)
Allocating user resources		Not required	Not required	Required
Hardware break function		2 points	2 points	2 points
Software break function	Internal ROM area	4 points	4 points	4 points
	RAM area	2000 points	2000 points	2000 points
Real-time RAM monitor function[Note 1]		Depending on the on-chip debug emulator product of partner	Available	Available
Dynamic memory modification (DMM) function[Note 2]		Depending on the on-chip debug emulator product of partner	Available	Available
Mask function		Reset, INTWDT	Reset, INTWDT	$\overline{\text{RESET}}$
ROM security function		10-byte ID code authentication	10-byte ID code authentication	10-byte ID code authentication
Hardware used		On-chip debug emulator product of partner	MINICUBE	MINICUBE2
Trace function		Available	Not supported	Not supported

Notes 1. This is a function which reads out memory contents during program execution.

2. This is a function which rewrites RAM contents during program execution.

26.1 Debugging Using DCU (Trace Function)

The DCU consists of three function units: an execution control unit (RCU) that realizes communication with JTAG and execution of debug processing, a trace control unit (TCU) that implements trace functions, and a trigger event unit (TEU) that implements event detection functions. On-chip debugging of the V850E/IH4-H can be executed by connecting an on-chip debug emulator of a partner.

Caution The debug function is supported by the V850E/IH4-H, but whether this function can be used or not depends on the debugger used.

26.1.1 Functional Outline

(1) Debug function

(a) Debug interface

This interface establishes communication with the host machine by using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO signals, via an on-chip debug emulator of partner. The communication specifications of JTAG are used for this interface. It does not support a boundary scan function.

(b) On-chip debugging

On-chip debugging can be performed if wiring and connectors for debugging are provided on the target system.

Connect an on-chip debug emulator of partner to the connector for debugging.

(c) Forced reset function

The V850E/IH4-H can be forcibly reset.

(d) Forced break function

Execution of the user program can be forcibly stopped (however, the handler of the illegal instruction code exception (first address: 00000060H) cannot be used).

(e) Debug monitor function

During debugging, a memory space for debugging, different from the user memory space, is used (background monitor format). The user program can be executed starting from any address.

While execution of the user program is stopped, the user resources (such as memory and I/O) can be read/written, and the user program can be downloaded.

(f) Mask function

(i) Non-maskable interrupt signal (INTWDT) and all maskable interrupt request signals can be masked.

(ii) When the debugger is connected, the $\overline{\text{RESET}}$ pin input on the target board is masked by default (the $\overline{\text{RESET}}$ pin input is masked when the debugger is started after power application to the V850E/IH4-H).

The $\overline{\text{RESET}}$ pin input can be unmasked from the debugger. If a signal is input to the $\overline{\text{RESET}}$ pin during debugging (during RUN execution), however, the following problems may occur.

- The break function may malfunction. If this happens, restart.
- Trace data may be illegal before and after $\overline{\text{RESET}}$ pin input. Recovery will occur after the $\overline{\text{RESET}}$ signal has been released.

(2) Trace function

(a) PC trace (branch trace) function

All branches (transition of processing) that occur during user program execution can be traced.

The trace sources can be selected from 12 types of branch sources that are classified by function, and PC trace can be started from execution of an instruction at any address, and the trace source can be changed.

Two trace start triggers are available.

(b) Data trace function

A data access issued by the CPU to any address can be traced in a range of 1 KB to 4 bytes.

Read or written data can be traced, and two data trace points are available.

However, a data access issued by the DMAC cannot be traced.

(c) Real-time trace mode

Branch and data access can be traced during real-time execution of the user program.

The trace packet of the trace source detected is stored in a trace buffer, and output from trace interface pins (TRCCLK, TRCDATA0 to TRCDATA3, and TRCEND) (some trace packets may not be traced if no more trace packets can be stored in the trace buffer).

(d) Full trace mode (non-real-time trace mode)

All branches and data accesses of the user program can be traced.

In the full trace mode, the pipeline of the CPU is temporarily held and instruction execution is stopped to secure the time of trace data output from trace interface pins, so that all trace packets can be correctly traced.

(3) Event function

(a) Instruction event detection function

Event detection (10 events) via size comparison by the execution PC and range event detection (up to four pairs with each pair consisting of two events) of the execution PC can be executed.

If an instruction event source is used as a break source, two breakpoints before execution of the instruction at which an event is detected and eight breakpoints after instruction execution can be detected.

(b) Access event detection function

Events can be detected as follows.

- Comparison of access addresses (4 addresses)
- Range of access address (up to two pairs with each pair consisting of two addresses)
- Match or mismatch of access data
- Data of specific bit by masking data
- Access size

An access event source is detected after access. If an access event source is used as a break source, a break occurs after several instructions have been executed after the instruction that issued the access that caused event detection.

(c) Sequential event detection function

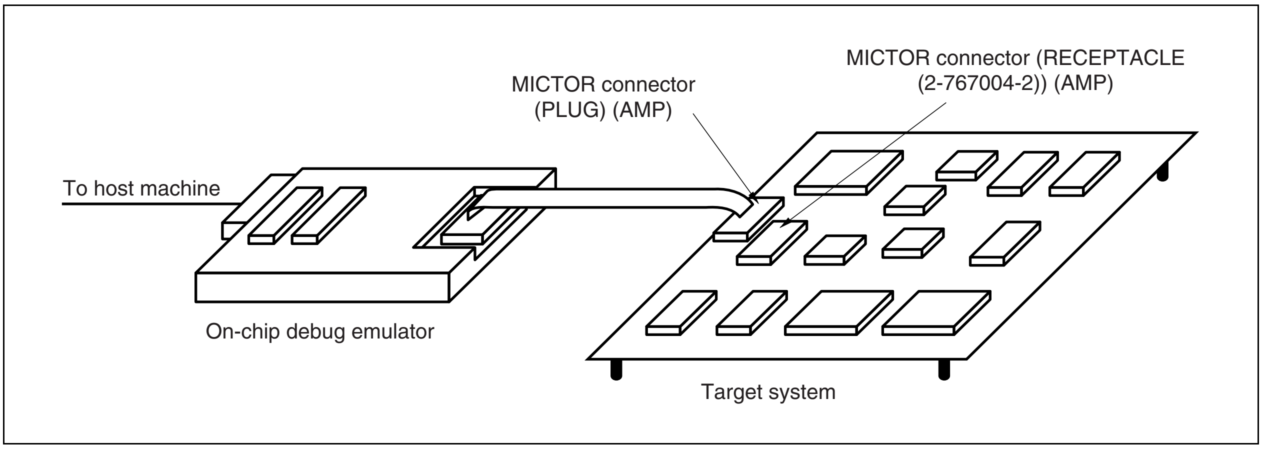
An event can be detected when up to four stages of events have successively occurred or an event that clears successive occurrence of events can be detected.

Sequential events can be counted by using a 12-bit pass counter.

26.1.2 Connection with on-chip debug emulator of partner

A connector for the emulator and a connection circuit must be provided on the target system.

Figure 26-1. Connecting On-Chip Debug Emulator of Partner


MICTOR connector (RECEPTACLE (2-767004-2)) (AMP)
MICTOR connector (PLUG) (AMP)
To host machine
On-chip debug emulator
Target system

(1) Emulator connector

The following table shows the pin functions of the emulator connector.

Table 26-2. Emulator Connector Pin Function

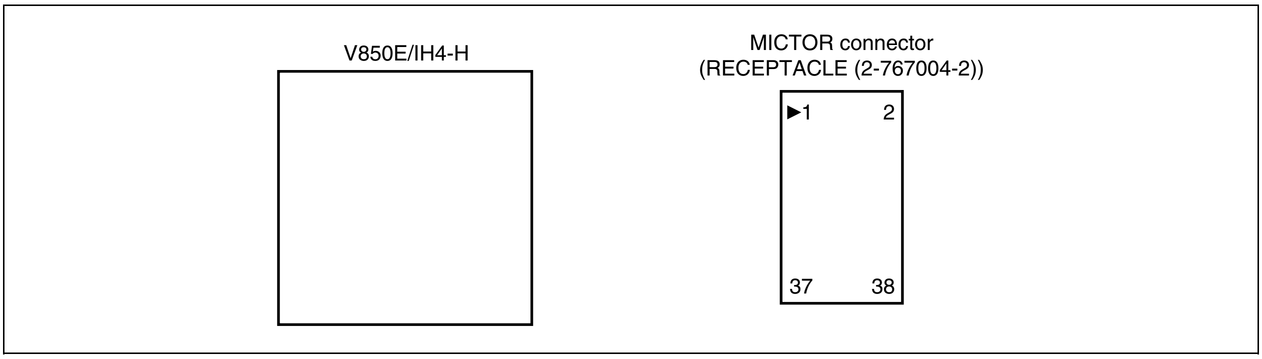
Pin No.	Pin Name	I/O Direction	Pin Function
1	GND	–	–
2	GND	–	–
3	DCK	V850E/IH4-H ← Emulator	Clock for debug serial interface (V850E/IH4-H ← Emulator)
4	V_{DD}	–	5 V (V850E/IH4-H → Emulator) (for monitoring power to target)
5	DMS	V850E/IH4-H ← Emulator	Transfer mode selection for debug serial interface (V850E/IH4-H ← Emulator)
6	$\overline{\text{DRST}}$	V850E/IH4-H ← Emulator	DCU reset (V850E/IH4-H ← Emulator)
7	DDI	V850E/IH4-H ← Emulator	Data for debug serial interface (V850E/IH4-H ← Emulator)
8	$\overline{\text{RESET}}$	V850E/IH4-H ← Emulator	System reset input signal (V850E/IH4-H ← Emulator)
9	DDO	V850E/IH4-H → Emulator	Data for debug serial interface (V850E/IH4-H ← Emulator)
10	FLMD0	V850E/IH4-H ← Emulator	Programming mode signal (V850E/IH4-H ← Emulator)
11	(Reserved 1)	–	(Leave this pin open)
12	PORT0_OUT	V850E/IH4-H ← Emulator	General-purpose control signal 0 (V850E/IH4-H ← Emulator)
13	(Reserved 2)	–	(Leave this pin open)
14	PORT0_IN	V850E/IH4-H → Emulator	General-purpose control signal 0 (V850E/IH4-H → Emulator)
15	(Reserved 3)	–	(Leave this pin open)
16	PORT1_IN	V850E/IH4-H → Emulator	General-purpose control signal 1 (V850E/IH4-H → Emulator)
17	TRCCLK	V850E/IH4-H → Emulator	Trace clock (V850E/IH4-H → Emulator)
18	PORT2_IN	V850E/IH4-H → Emulator	General-purpose control signal 2 (V850E/IH4-H → Emulator)
19	TRCEND	V850E/IH4-H → Emulator	Trace data end (V850E/IH4-H → Emulator)
20	TRCCE	V850E/IH4-H → Emulator	Trace packet compression enable signal (V850E/IH4-H → Emulator)
21	TRCDATA0	V850E/IH4-H → Emulator	Trace data 0 (V850E/IH4-H → Emulator)
22	TRCDATA8	V850E/IH4-H → Emulator	Trace data 8 (V850E/IH4-H → Emulator)
23	TRCDATA1	V850E/IH4-H → Emulator	Trace data 1 (V850E/IH4-H → Emulator)
24	TRCDATA9	V850E/IH4-H → Emulator	Trace data 9 (V850E/IH4-H → Emulator)
25	TRCDATA2	V850E/IH4-H → Emulator	Trace data 2 (V850E/IH4-H → Emulator)
26	TRCDATA10	V850E/IH4-H → Emulator	Trace data 10 (V850E/IH4-H → Emulator)
27	TRCDATA3	V850E/IH4-H → Emulator	Trace data 3 (V850E/IH4-H → Emulator)
28	TRCDATA11	V850E/IH4-H → Emulator	Trace data 11 (V850E/IH4-H → Emulator)
29	TRCDATA4	V850E/IH4-H → Emulator	Trace data 4 (V850E/IH4-H → Emulator)
30	TRCDATA12	V850E/IH4-H → Emulator	Trace data 12 (V850E/IH4-H → Emulator)
31	TRCDATA5	V850E/IH4-H → Emulator	Trace data 5 (V850E/IH4-H → Emulator)
32	TRCDATA13	V850E/IH4-H → Emulator	Trace data 13 (V850E/IH4-H → Emulator)
33	TRCDATA6	V850E/IH4-H → Emulator	Trace data 6 (V850E/IH4-H → Emulator)
34	TRCDATA14	V850E/IH4-H → Emulator	Trace data 14 (V850E/IH4-H → Emulator)
35	TRCDATA7	V850E/IH4-H → Emulator	Trace data 7 (V850E/IH4-H → Emulator)
36	TRCDATA15	V850E/IH4-H → Emulator	Trace data 15 (V850E/IH4-H → Emulator)
37	GND	–	–
38	GND	–	–

Remark **Cautions** are given on the next page.

Cautions 1. The connection of pins not supported by the V850E/IH4-H depends on the emulator used.

2. The pattern on the target board must satisfy the following conditions to support high-speed interfacing.

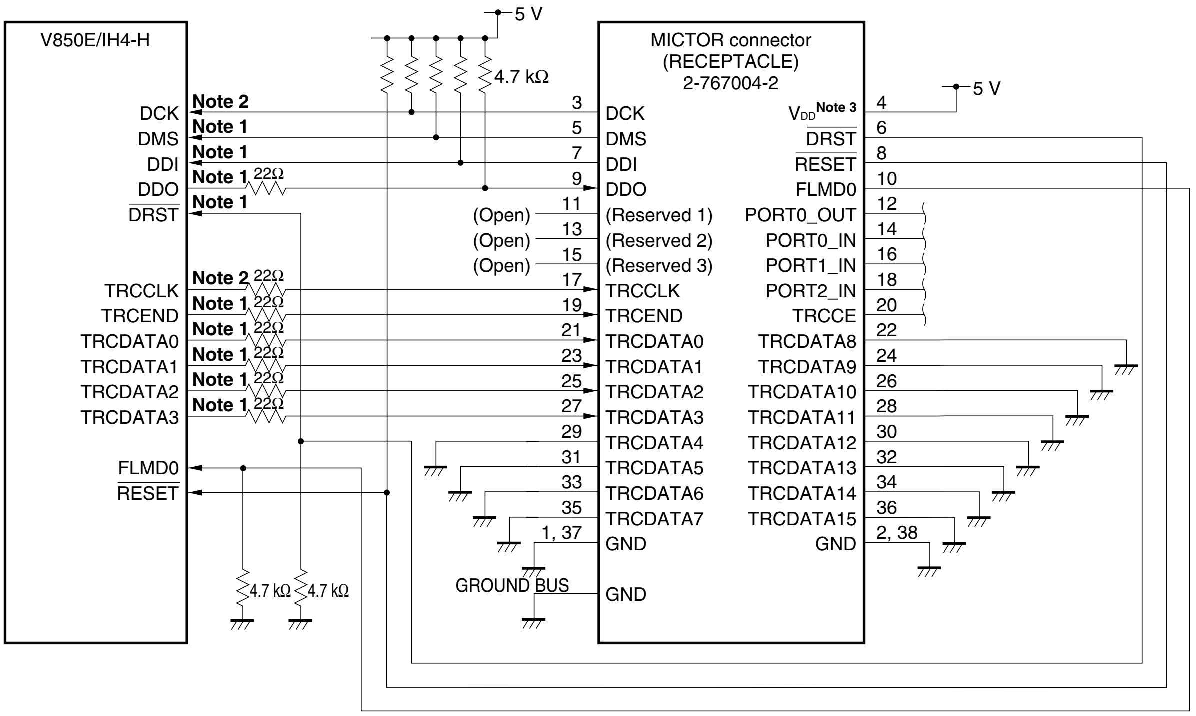
- **Lay out the pattern with the odd number of pins facing the device (V850E/IH4-H).**
- **Keep the pattern length to within 1.97 inches (50 mm).**
- **Shield the clock signal with GND.**



(2) Recommended circuit example

The following figure shows an example of the recommended circuit of the emulator connector (on the target system side).

Figure 26-2. Example of Recommended Emulator Connection Circuit



Notes 1. Keep the pattern length to within 1.97 inches (50 mm).

2. Shield the DCK and TRCCLK signals by GND.

3. For detecting power to the target board

Caution The recommended circuit example shown above assumes that a 5 V interface is used.

26.2 Debugging Using DCU (No Trace Function)

The program can be debugged by using the debug interface pins ($\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO) and connecting an on-chip debug simulator (MINICUBE).

26.2.1 Circuit connection examples

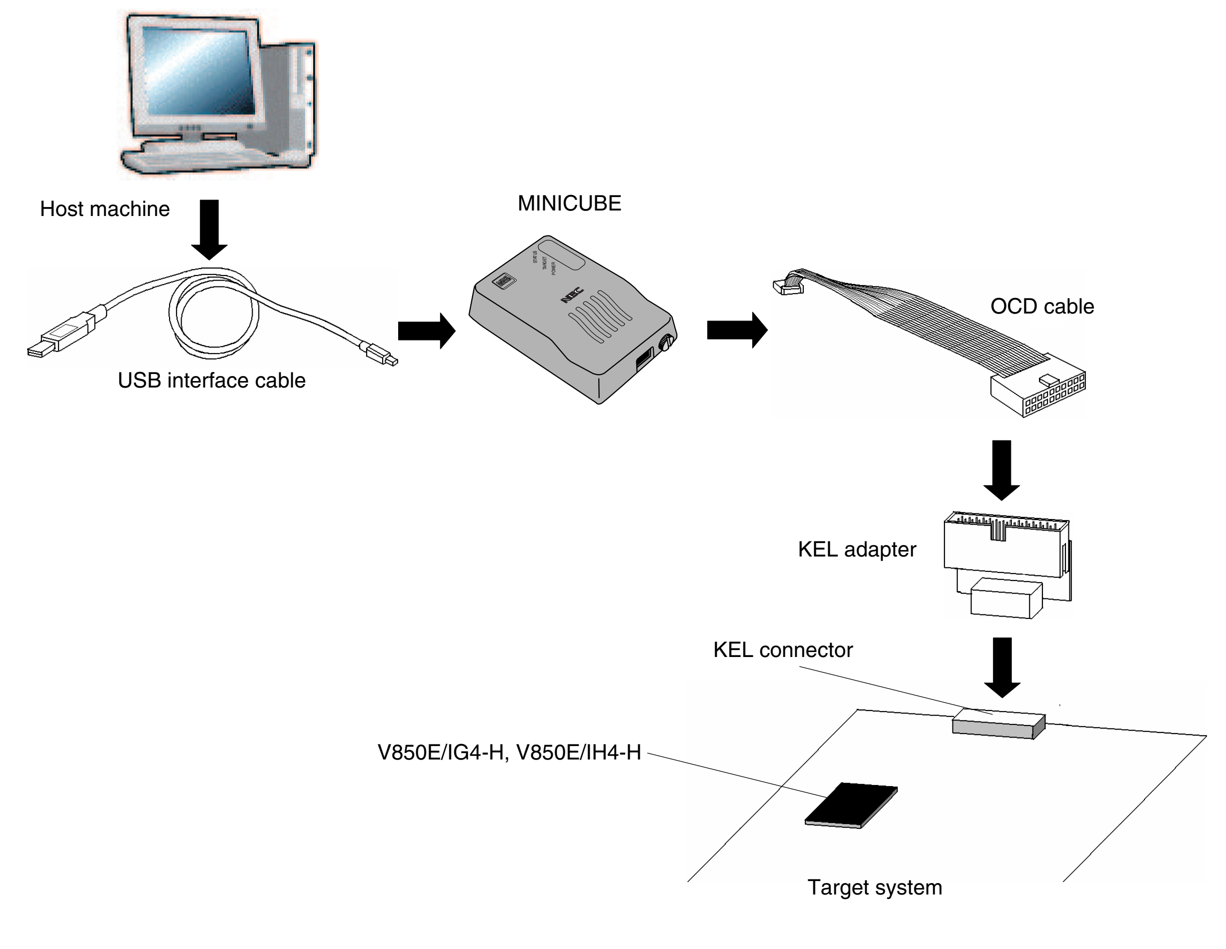
When the MINICUBE is used, use of the following KEL connector is recommended.

○ Part number

- 8830E-026-170S: Straight type
- 8830E-026-170L: Right-angle type

It is necessary to mount an emulator and circuit for connection on the target system.

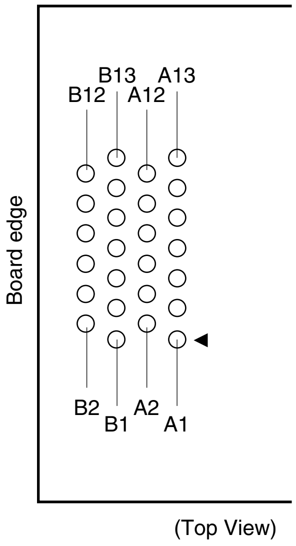
Figure 26-3. Connection Example of On-Chip Debug Emulator (MINICUBE)



(1) Pin configuration

The following figure shows the pin configuration of the emulator connector (on the target system side).

Figure 26-4. Pin Configuration of Emulator Connector (on Target System Side)



Caution Design the board based on the dimensions of the connector when actually mounting the connector on the board.

(2) Pin functions

The following table shows the pin functions of the emulator connector (on the target system side).

Table 26-3. Pin Functions of Emulator Connector (on Target System Side)

Pin No.	Pin Name	I/O	Pin Function
A1	(Reserved 1)	–	(Connect to GND)
A2	(Reserved 2)	–	(Connect to GND)
A3	(Reserved 3)	–	(Connect to GND)
A4	(Reserved 4)	–	(Connect to GND)
A5	(Reserved 5)	–	(Connect to GND)
A6	(Reserved 6)	–	(Connect to GND)
A7	DDI	Output	Data output for debug serial interface
A8	DCK	Output	Clock output for debug serial interface
A9	DMS	Output	Transfer mode select output for debug serial interface
A10	DDO	Input	Data input for debug serial interface
A11	$\overline{\text{DRST}}$	Output	DCU reset output
A12	(Reserved 7)	–	(Leave open)
A13	FLMD0	Output	Control signal for flash memory downloading
B1	GND	–	–
B2	GND	–	–
B3	GND	–	–
B4	GND	–	–
B5	GND	–	–
B6	GND	–	–
B7	GND	–	–
B8	GND	–	–
B9	GND	–	–
B10	GND	–	–
B11	PORT0_IN	–	(Connect to GND)
B12	PORT1_IN	–	(Connect to GND)
B13	V_{DD}	–	5 V input (for monitoring power application to target)

Cautions 1. The connection of the pins not supported in the V850E/IG4-H and V850E/IH4-H depends on the emulator used.

2. The pattern on the target board must satisfy the following conditions.

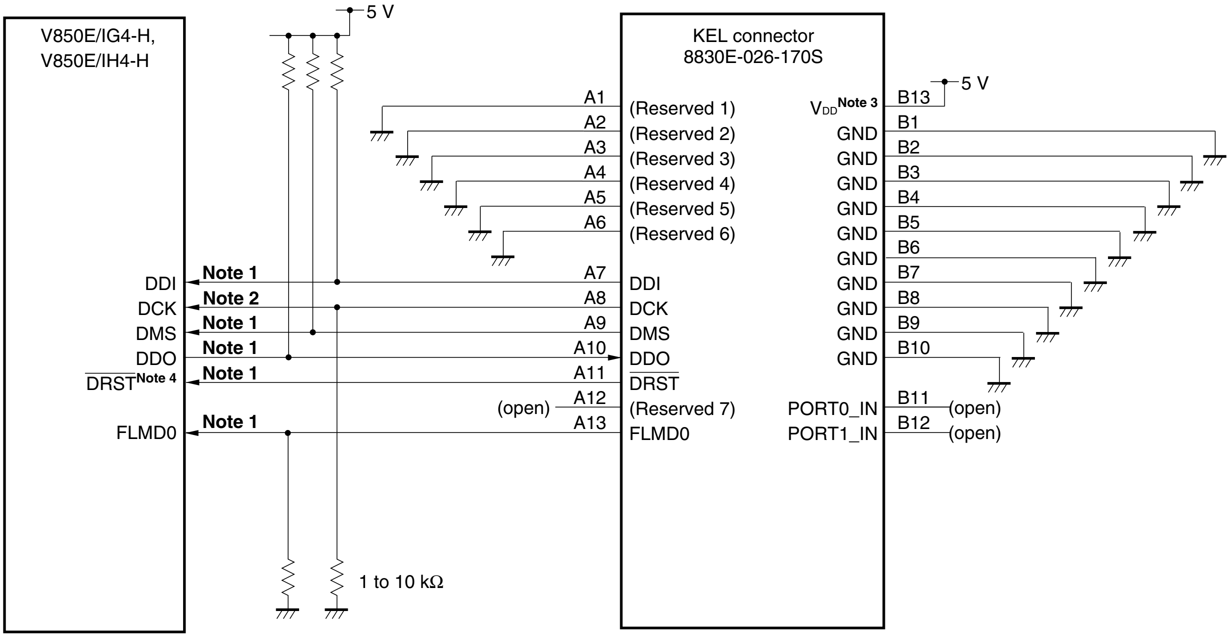
- **Keep the pattern length to within 100 mm.**
- **Shield the clock signal with GND.**

Remark Input/output is as viewed from the emulator side.

(3) Recommended circuit example

The following figure shows an example of the recommended circuit of the emulator connector (on the target system side).

Figure 26-5. Example of Recommended Connection of Emulator



Notes 1. Keep the pattern length to within 100 mm.

2. Shield the DCK signal with GND.

3. For detecting power supply to the target board.

4. When $\overline{\text{DRST}}$ pin is high level: On-chip debug mode
When $\overline{\text{DRST}}$ pin is low level: Normal operation mode
The $\overline{\text{DRST}}$ pin is internally pulled down in the V850E/IG4-H and V850E/IH4-H.

Caution The DDO signal is 5 V output, and the input level of the DDI, DCK, DMS, and $\overline{\text{DRST}}$ signals is TTL level.

26.2.2 Interface signals

The interface signals on the V850E/IG4-H or V850E/IH4-H side are described below.

(1) $\overline{\mathrm{DRST}}$

This is a reset input signal for the on-chip debug unit. It is a negative-logic signal that asynchronously initializes the debug control unit (DCU).

MINICUBE changes the level of the $\overline{\mathrm{DRST}}$ signal from low to high for output and starts the on-chip debug unit of the V850E/IG4-H and V850E/IH4-H when it detects VDD of the target system after the integrated debugger is started. If VDD is not detected from the target system, the output signals ($\overline{\mathrm{DRST}}$, DCK, DMS, DDI, and FLMD0 pins) from the MINICUBE go into a high-impedance state.

When the $\overline{\mathrm{DRST}}$ signal goes high, a reset signal is also generated in the V850E/IG4-H and V850E/IH4-H.

When starting debugging by starting the integrated debugger, a reset signal is always generated.

(2) DCK

This is a clock input signal. It supplies a 20 MHz clock from MINICUBE. In the on-chip debug unit, the DMS and DDI signals are sampled at the rising edge of the DCK signal, and the data DDO is output at its falling edge.

(3) DMS

This is a transfer mode select signal. The transfer status in the debug unit changes depending on the level of the DMS signal.

(4) DDI

This is a data input signal. It is sampled in the on-chip debug unit at the rising edge of DCK.

(5) DDO

This is a data output signal. It is output from the on-chip debug unit at the falling edge of the DCK signal.

(6) FLMD0

The flash self programming function is used for the function to download data to the flash memory via the integrated debugger. During flash self programming, the FLMD0 pin must be kept high. In addition, connect a pull-down resistor to the FLMD0 pin.

The FLMD0 pin can be controlled in either of the following two ways.

<1> To control from MINICUBE

Connect the FLMD0 signal of MINICUBE to the FLMD0 pin of the V850E/IG4-H and V850E/IH4-H.

In the normal mode, nothing is driven by MINICUBE (high impedance).

During a break, MINICUBE raises the FLMD0 pin to the high level when the download function of the integrated debugger is executed.

<2> To control from port

Connect any port of the device to the FLMD0 pin of the V850E/IG4-H and V850E/IH4-H.

The same port as the one used by the user program to realize the flash self programming function may be used.

On the console of the integrated debugger, make a setting to raise the port pin to high level before executing the download function, or lower the port pin after executing the download function.

For details, refer to the **ID850QB Ver. 3.40 Integrated Debugger Operation User's Manual (U18604E)**.

26.2.3 Maskable functions

Reset and INTWDT signals can be masked.

The maskable functions with the debugger (ID850QB) and the corresponding functions are shown below.

Table 26-4. Maskable Functions

Maskable Functions with Debugger (ID850QB)	Corresponding Function of V850E/IG4-H, V850E/IH4-H
NMI0	Non-maskable interrupt request signal (INTWDT) generation
NMI1	×
NMI2	×
STOP	×
HOLD	×
RESET	$\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIRES) generation by low-voltage detector (LVI), reset signal (POCRES) generation by power-on-clear circuit (POC)

26.2.4 Cautions

(1) If a reset signal is input (from the target system or due to the execution of an internal reset) while the program is running, the software breaks specified for the on-chip flash memory area will no longer occur. Use hardware breaks to avoid this problem. The disabled software breaks can be enabled again by generating a forcible break or a hardware break.

(2) Pin reset during a break is masked and the CPU and peripheral I/O are not reset. If pin reset or internal reset is generated as soon as the flash memory is rewritten by DMA or read by the RAM monitor function while the user program is being executed, the CPU and peripheral I/O may not be correctly reset.

(3) In the on-chip debug mode, the DDO pin is forcibly set to the high-level output.

(4) The flash memory of the device used in debugging is rewritten during debugging, so the number of flash memory rewrites cannot be guaranteed. Therefore, do not use the device used in debugging for a mass production product.

(5) Because the DDI and DCK pins function alternately as the CSIF0 I/O pins (SIF0, $\overline{\text{SCKF0}}$), UARTA0 input pin (RXDA0), and TAA output pin (TOA00, TOA10), CSIF0, UARTA0, and TAA0 cannot be used while the on-chip debug function is being used.

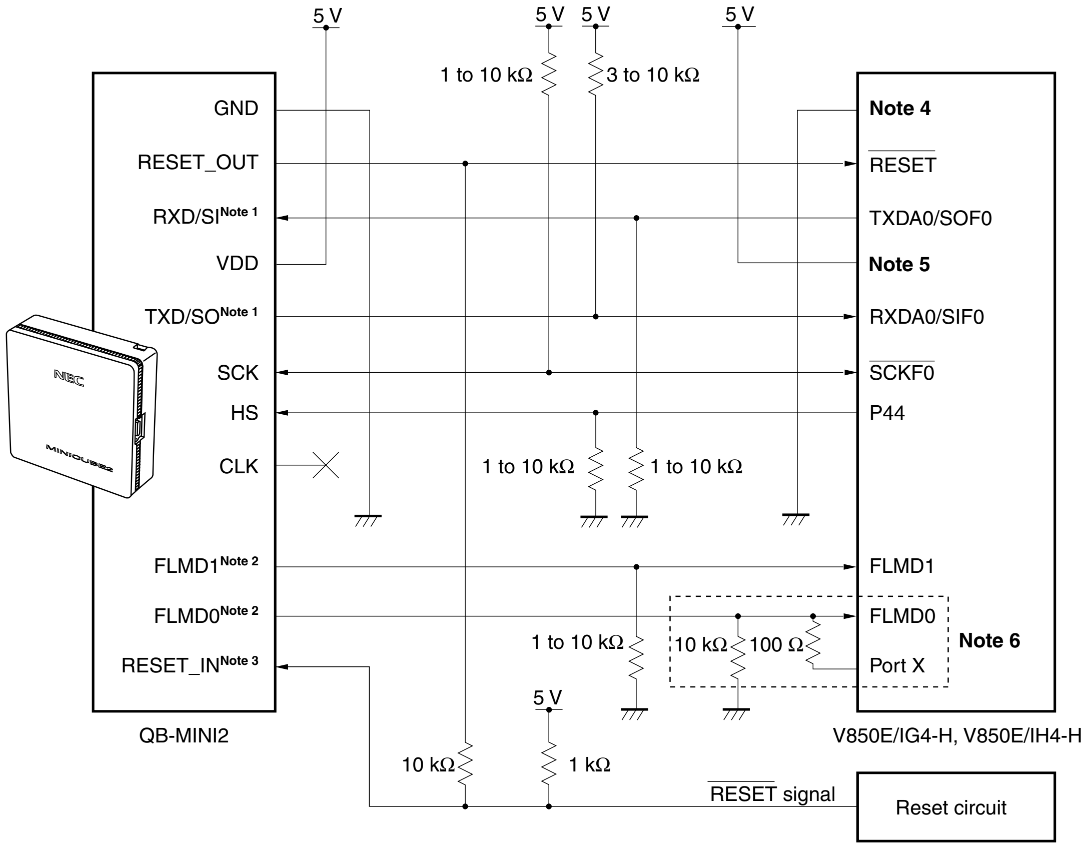
(6) When the on-chip debug function is used, the clock generator and PLL continue operating even if the STOP mode is set.

26.3 Debugging Without Using DCU

The following describes how to implement an on-chip debug function using MINICUBE2 with the UARTA0 pins (RXDA0, TXDA0) or CSIF0 pins (SIF0, SOF0, $\overline{\text{SCKF0}}$, HS (P44)) as debug interfaces, without using the DCU.

26.3.1 Circuit connection examples

Figure 26-6. Circuit Connection Example When UARTA0/CSIF0 Is Used for Communication Interface



Notes 1. Connect TXDA0/SOF0 (transmission side) of the V850E/IG4-H and V850E/IH4-H to RXD/SI (reception side) of the target connector, and TXD/SO (transmission side) of the target connector to RXDA0/SIF0 (reception side) of the V850E/IG4-H and V850E/IH4-H.

2. The V850E/IG4-H or V850E/IH4-H-side pin connected to this pin (FLMD0, FLMD1) can be used as an alternate-function pin other than while the memory is rewritten during a break in debugging, because this pin is in a Hi-Z state.

3. This connection is designed assuming that the $\overline{\text{RESET}}$ signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less).

4. EV$_{SS0}$, EV$_{SS1}$, EV$_{SS2}$, EV$_{SS3}$ (V850E/IH4-H only), EV$_{SS4}$, V$_{SS0}$, V$_{SS1}$, V$_{SS2}$, AV$_{SS0}$, AV$_{SS1}$, AV$_{SS2}$

5. EV$_{DD0}$, EV$_{DD1}$, EV$_{DD2}$, EV$_{DD3}$ (V850E/IH4-H only), FV$_{DD}$ (V850E/IH4-H only), V$_{DD0}$, V$_{DD1}$, V$_{DD2}$, AV$_{DD0}$, AV$_{DD1}$, AV$_{DD2}$

6. The circuit enclosed by broken lines is designed for flash self programming, which controls the FLMD0 pin via ports. Use the port for inputting or outputting the high level. When flash self programming is not performed, a pull-down resistance for the FLMD0 pin can be within 1 to 10 kΩ.

Remark See **Table 26-5** for pins used when UARTA0 or CSIF0 is used for communication interface.

Table 26-5. Wiring Between V850E/IG4-H or V850E/IH4-H and MINICUBE2 (1/2)

Pins Connected to MINICUBE2 (QB-MINI2)			When UARTA0 Used		
Signal Name	I/O	Pin Function	Pin Name	Pin No.	
				V850E/IG4-H	V850E/IH4-H
				GC	GF
SI/RxD	Input	Pin to receive commands and data from V850E/IG4-H and V850E/IH4-H	TXDA0	47	97
SO/TxD	Output	Pin to transmit commands and data to V850E/IG4-H and V850E/IH4-H	RXDA0	46	96
SCK	Output	Clock output pin for 3-wire serial communication	Not needed	Not needed	Not needed
CLK	Output	Clock output pin to V850E/IG4-H and V850E/IH4-H	Not needed	Not needed	Not needed
RESET_OUT	Output	Reset output pin to V850E/IG4-H and V850E/IH4-H	$\overline{\text{RESET}}$	39	82
FLMD0	Output	Output pin to set V850E/IG4-H and V850E/IH4-H to debug mode or programming mode	FLMD0	42	86
FLMD1	Output	Output pin to set programming mode	FLMD1	76	1
HS	Input	Handshake signal for CSI0 + HS communication	Not needed	Not needed	Not needed
GND	–	Ground	V_{SS0}	38	81
			V_{SS1}	64	117
			V_{SS2}	91	28
			AV_{SS0}	5	43
			AV_{SS1}	10	48
			AV_{SS2}	27	66
			EV_{SS0}	41	85
			EV_{SS1}	63	116
			EV_{SS2}	100	38
			EV_{SS3}	–	8
			EV_{SS4}	31	74
RESET_IN	Input	Reset input pin on the target system			

Table 26-5. Wiring Between V850E/IG4-H or V850E/IH4-H and MINICUBE2 (2/2)

Pins Connected to MINICUBE2 (QB-MINI2)			When CSIF0-HS Used		
Signal Name	I/O	Pin Function	Pin Name	Pin No.	
				V850E/IG4-H	V850E/IH4-H
				GC	GF
SI/RxD	Input	Pin to receive commands and data from V850E/IG4-H and V850E/IH4-H	SOF0	47	97
SO/TxD	Output	Pin to transmit commands and data to V850E/IG4-H and V850E/IH4-H	SIF0	46	96
SCK	Output	Clock output pin for 3-wire serial communication	$\overline{\text{SCKF0}}$	48	98
CLK	Output	Clock output pin to V850E/IG4-H and V850E/IH4-H	Not needed	Not needed	Not needed
RESET_OUT	Output	Reset output pin to V850E/IG4-H and V850E/IH4-H	$\overline{\text{RESET}}$	39	82
FLMD0	Output	Output pin to set V850E/IG4-H and V850E/IH4-H to debug mode or programming mode	FLMD0	42	86
FLMD1	Output	Output pin to set programming mode	FLMD1	76	1
HS	Input	Handshake signal for CSI0 + HS communication	P44	50	100
GND	–	Ground	V_{SS0}	38	81
			V_{SS1}	64	117
			V_{SS2}	91	28
			AV_{SS0}	5	43
			AV_{SS1}	10	48
			AV_{SS2}	27	66
			EV_{SS0}	41	85
			EV_{SS1}	63	116
			EV_{SS2}	100	38
			EV_{SS3}	–	8
			EV_{SS4}	31	74
RESET_IN	Input	Reset input pin on the target system			

26.3.2 Maskable functions

Reset signal can only be masked.

The maskable functions with the debugger (ID850QB) and the corresponding functions are shown below.

Table 26-6. Maskable Functions

Maskable Functions with Debugger (ID850QB)	Corresponding Function of V850E/IG4-H, V850E/IH4-H
NMI0	×
NMI1	×
NMI2	×
STOP	×
HOLD	×
RESET	Reset signal generation by $\overline{\text{RESET}}$ pin input

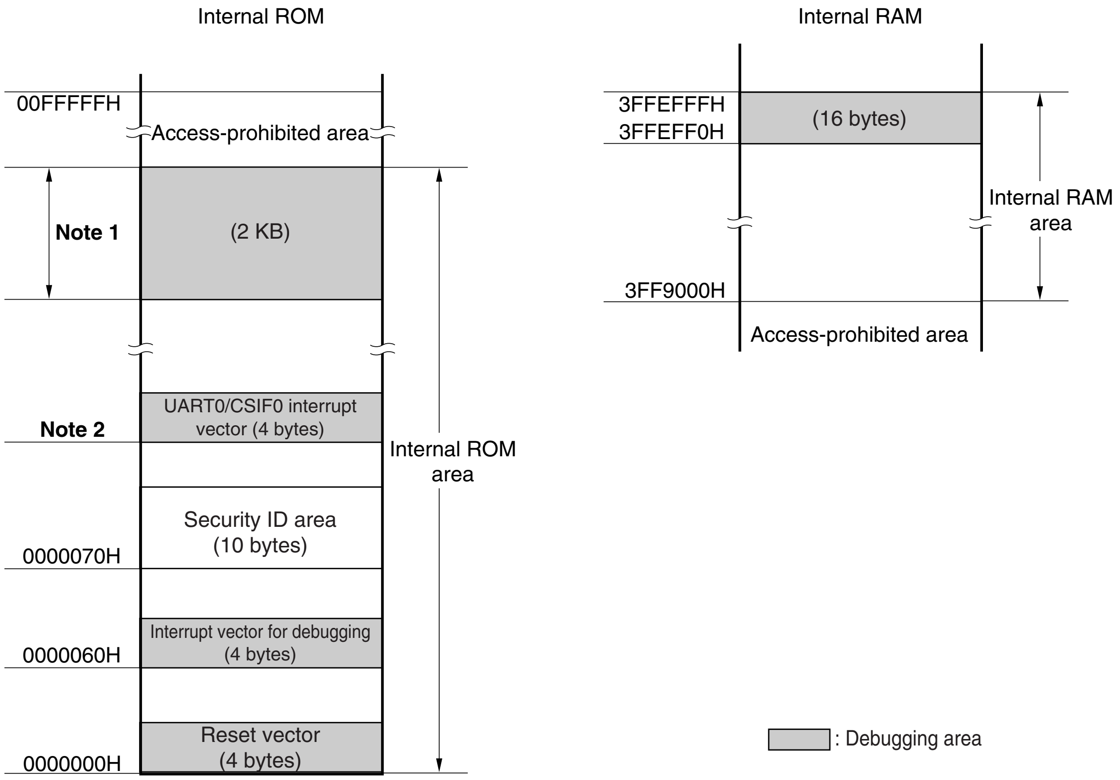
26.3.3 Securing of user resources

The user must prepare the following to perform communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H and implement each debug function. These items need to be set in the user program or using the compiler options.

(1) Securement of memory space

The shaded portions in Figure 26-7 are the areas reserved for placing the debug monitor program, so user programs and data cannot be allocated in these spaces. These spaces must be secured so as not to be used by the user program.

Figure 26-7. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address values vary depending on the product.

	Internal ROM size	Debugging area
μPD70F3919 (V850E/IG4-H)	256 KB	003F800H to 003FFFFH
μPD70F3922 (V850E/IH4-H)		
μPD70F3920 (V850E/IG4-H)	384 KB	005F800H to 005FFFFH
μPD70F3923 (V850E/IH4-H)		
μPD70F3921 (V850E/IG4-H)	480 KB	0077800H to 0077FFFH
μPD70F3924 (V850E/IH4-H)		

2. Start address values when UARTA0 and CSIF0 are used are as follows.

Target serial interface	Interrupt name	Start address
UARTA0	INTUA0RE	000004E0H
	INTUA0R	000004F0H
	INTUA0T	00000500H
CSIF0	INTCF0RE	00000510H
	INTCF0R	00000520H
	INTCF0T	00000530H

- Security ID setting

  The ID code must be embedded in the area between 0000070H and 0000079H in Figure 26-7, to prevent the memory from being read by an unauthorized person. For details, see **26.4 ROM Security Function**.

(2) Reset vector

A reset vector includes the jump instruction for the debug monitor program.

[How to secure areas]

It is not necessary to secure this area intentionally. When downloading a program, however, the debugger rewrites the reset vector in accordance with the following cases. If the rewritten pattern does not match the following cases, the debugger generates an error (F0c34 when using the ID850QB).

(a) When two nop instructions are placed in succession from address 0

Before rewriting		After rewriting
0x0 nop	→	Jumps to debug monitor program at 0x0
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) When two 0xFFFF are successively placed from address 0 (already erased device)

Before rewriting		After rewriting
0x0 0xFFFF	→	Jumps to debug monitor program at 0x0
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) The *jr* instruction is placed at address 0 (when using CA850)

Before rewriting		After rewriting
0x0 jr disp22	→	Jumps to debug monitor program at 0x0
		0x4 jr disp22 - 4

(d) *mov32* and *jmp* are placed in succession from address 0 (when using IAR compiler ICCV850)

Before rewriting		After rewriting
0x0 mov imm32,reg1	→	Jumps to debug monitor program at 0x0
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

(e) The jump instruction for the debug monitor program is placed at address 0

Before rewriting		After rewriting
Jumps to debug monitor program at 0x0	→	No change

(3) Securement of area for debug monitor program

The shaded portions in Figure 26-7 are the areas where the debug monitor program is allocated. The monitor program performs initialization processing for debug communication interface and RUN or break processing for the CPU. The internal ROM area must be filled with 0xFF. This area must not be rewritten by the user program.

[How to secure areas]
It is not necessarily required to secure this area if the user program does not use this area.
To avoid problems that may occur during the debugger startup, however, it is recommended to secure this area in advance, using the compiler.
The following shows examples for securing the area, using the Renesas Electronics compiler CA850. Add the assemble source file and link directive code, as shown below.

- Assemble source (Add the following code as an assemble source file.)

```
-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space   0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space   4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "INTCF0RE"
.space   4, 0xff
.section "INTCF0R"
.space   4, 0xff
.section "INTCF0T"
.space   4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm   monitorramsym, 16, 4;    -- defines symbol monitorramsym
```

- Link directive (Add the following code to the link directive file.)
  The following shows an example when the internal ROM has 256 KB (end address is 003FFFFH) and internal RAM has 24 KB (end address is 3FFEFFFH).

```
MROMSEG       : !LOAD ?R V0x03f800{
              MonitorROM    = $PROGBITS   ?A MonitorROM;
};

MRAMSEG       : !LOAD ?RW V0x03ffeff0{
              MonitorRAM    = $NOBITS     ?AW MonitorRAM;
};
```

(4) Securement of communication serial interface

UARTA0 or CSIF0 is used for communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H. The settings related to the serial interface modes are performed by the debug monitor program, but if the setting is changed by the user program, a communication error may occur.

To prevent such a problem from occurring, communication serial interface must be secured in the user program.

[How to secure communication serial interface]

- Serial interface registers
  Do not set the registers related to UARTA0 and CSIF0 in the user program.

- Interrupt mask register
  When UARTA0 is used, do not mask the reception end interrupt (INTUA0R). When CSIF0 is used, do not mask the reception end interrupt (INTCF0R).

(a) When UARTA0 is used

	7	6	5	4	3	2	1	0
UA0RIC	×	0	×	×	×	×	×	×

(b) When CSIF0 is used

	7	6	5	4	3	2	1	0
CF0RIC	×	0	×	×	×	×	×	×

Remark ×: 0 or 1

- Port registers when UARTA0 is used
  When UARTA0 is used for communication, port registers are set to make the TXDA0 and RXDA0 pins valid by the debug monitor program. Do not change the following register settings with the user program during debugging. (The same value can be overwritten.)

	7	6	5	4	3	2	1	0
PFCE4	0	0	0	0	0	×	0	0

	7	6	5	4	3	2	1	0
PFC4	0	0	0	×	×	0	1	1

	7	6	5	4	3	2	1	0
PMC4	0	0	0	×	×	×	1	1

Remark ×: 0 or 1

- Port registers when CSIF0 is used

  When CSIF0 is used, port registers are set to make the SIF0, SOF0, $\overline{\text{SCKF0}}$, and HS (P44) pins valid by the debug monitor program. Do not change the following register settings with the user program during debugging. (The same value can be overwritten.)

(a) SIF0, SOF0, and $\overline{\text{SCKF0}}$ settings

	7	6	5	4	3	2	1	0
PFCE4	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
PFC4	0	0	0	×	×	0	0	0

	7	6	5	4	3	2	1	0
PMC4	0	0	0	×	×	1	1	1

(b) HS (P44 pin) settings

	7	6	5	4	3	2	1	0
PMC4	0	0	0	0	×	×	×	×

	7	6	5	4	3	2	1	0
PM4	0	0	0	0	×	×	×	×

	7	6	5	4	3	2	1	0
P4	0	0	0	**Note**	×	×	×	×

Note Writing to this bit is prohibited.

The values corresponding to the HS pin are changed by the monitor program according to the debugger status. To perform port register settings in 8-bit units, the user program can usually use read-modify-write. If an interrupt for debugging occurs before writing, however, an unexpected operation may be performed.

Remark ×: 0 or 1

26.3.4 Cautions

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed. Moreover, do not embed the debug monitor program into mass-produced products.

(2) When breaks cannot be executed

Forced breaks cannot be executed if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H is UARTA0, and the peripheral clock has been stopped

(3) When pseudo real-time RAM monitor (RRM) function and DMM function do not operate

The pseudo RRM function and DMM function do not operate if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H is UARTA0, and the peripheral clock has been stopped
- Mode for communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H is UARTA0, and a clock different from the one specified in the debugger is used for communication

(4) Standby release with pseudo RRM and DMM functions enabled

The standby mode is released by the pseudo RRM function and DMM function if one of the following conditions is satisfied.

- Mode for communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H is CSIF0
- Mode for communication between MINICUBE2 and the V850E/IG4-H or V850E/IH4-H is UARTA0, and the peripheral clock has not stopped.

(5) Writing to peripheral I/O registers that requires a specific sequence, using DMM function

Peripheral I/O registers that requires a specific sequence cannot be written with the DMM function.

(6) Flash self programming

If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally.

26.4 ROM Security Function

26.4.1 Security ID

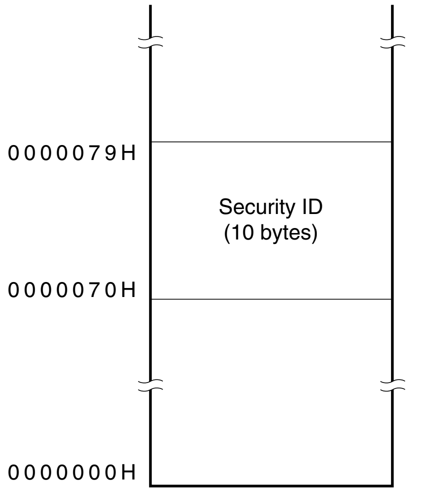
The flash memory versions of the V850E/IG4-H and V850E/IH4-H perform authentication using a 10-byte ID code to prevent the contents of the flash memory from being read by an unauthorized person during on-chip debugging by the on-chip debug emulator.

Set the ID code in the 10-byte on-chip flash memory area from 0000070H to 0000079H to allow the debugger perform ID authentication.

If the IDs match, the security is released and reading flash memory and using the on-chip debug emulator are enabled.

- Set the 10-byte ID code to 0000070H to 0000079H.
- Bit 7 of 0000079H is the on-chip debug emulator enable flag.
  (0: Disable, 1: Enable)
- When the on-chip debug emulator is started, the debugger requests ID input. When the ID code input on the debugger and the ID code set in 0000070H to 0000079H match, the debugger starts.
- Debugging cannot be performed if the on-chip debug emulator enable flag is 0, even if the ID codes match.

Figure 26-8. Security ID Area



Caution After the flash memory is erased, 1 is written to the entire area.

26.4.2 Setting

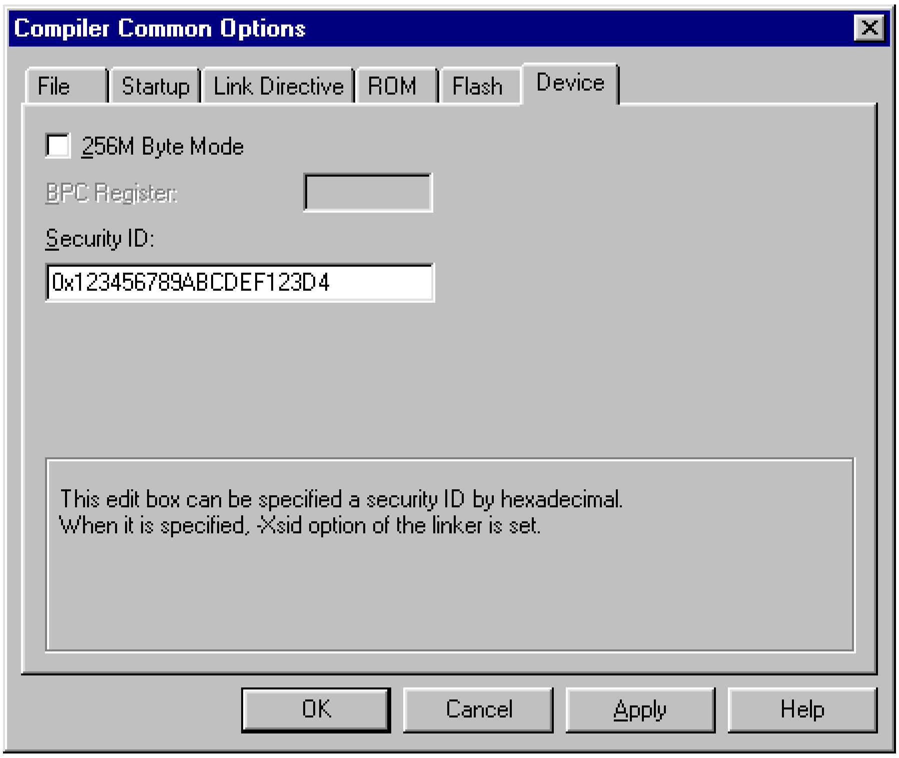
The following shows how to set the ID code as shown in Table 26-7.

When the ID code is set as shown in Table 26-7, the ID code input in the configuration dialog box of the ID850QB is “123456789ABCDEF123D4” (the ID code is case-insensitive).

Table 26-7. ID Code

Address	Value
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0XF1
0x78	0x23
0x79	0xD4

The ID code can be specified in the Compiler Common Options dialog box in PM+ if a device file that supports CA850 Ver. 2.60 and later and the security ID is used.



□ 256M Byte Mode

BPC Register:

Security ID: 0x123456789ABCDEF123D4

This edit box can be specified a security ID by hexadecimal.
When it is specified, -Xsid option of the linker is set.

CHAPTER 27 FLASH MEMORY

The V850E/IG4-H and V850E/IH4-H have a 256 KB, 384 KB, or 480 KB on-chip flash memory.

- μPD70F3919 (V850E/IG4-H), 70F3922 (V850E/IH4-H): 256 KB on-chip flash memory version
- μPD70F3920 (V850E/IG4-H), 70F3923 (V850E/IH4-H): 384 KB on-chip flash memory version
- μPD70F3921 (V850E/IG4-H), 70F3924 (V850E/IH4-H): 480 KB on-chip flash memory version

Flash memory can be rewritten with the flash memory programmer or using the self programming mode.

Writing to the flash memory programmer can be performed with the flash memory programmer that is connected to the target system.

Writing in the self programming mode can be performed with an application program, without using the flash memory programmer.

Flash memory versions are commonly used in the following development environments and mass production applications.

○ For altering software after the V850E/IG4-H and V850E/IH4-H is soldered onto the target system.
○ For differentiating software according to the specification in small scale production of various models.
○ For data adjustment when starting mass production.

27.1 Features

○ All area batch erase or erase in block units (4 KB)
○ Communication through serial interface from the flash memory programmer
○ Erase/write voltage: Erase/write is possible with a single power supply
○ On-board programming
○ Flash memory self programming possible
○ Secure rewriting of entire flash memory area by self programming using boot swap function
○ Rewriting method
  - Rewriting by communication with flash memory programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by user program (self programming)

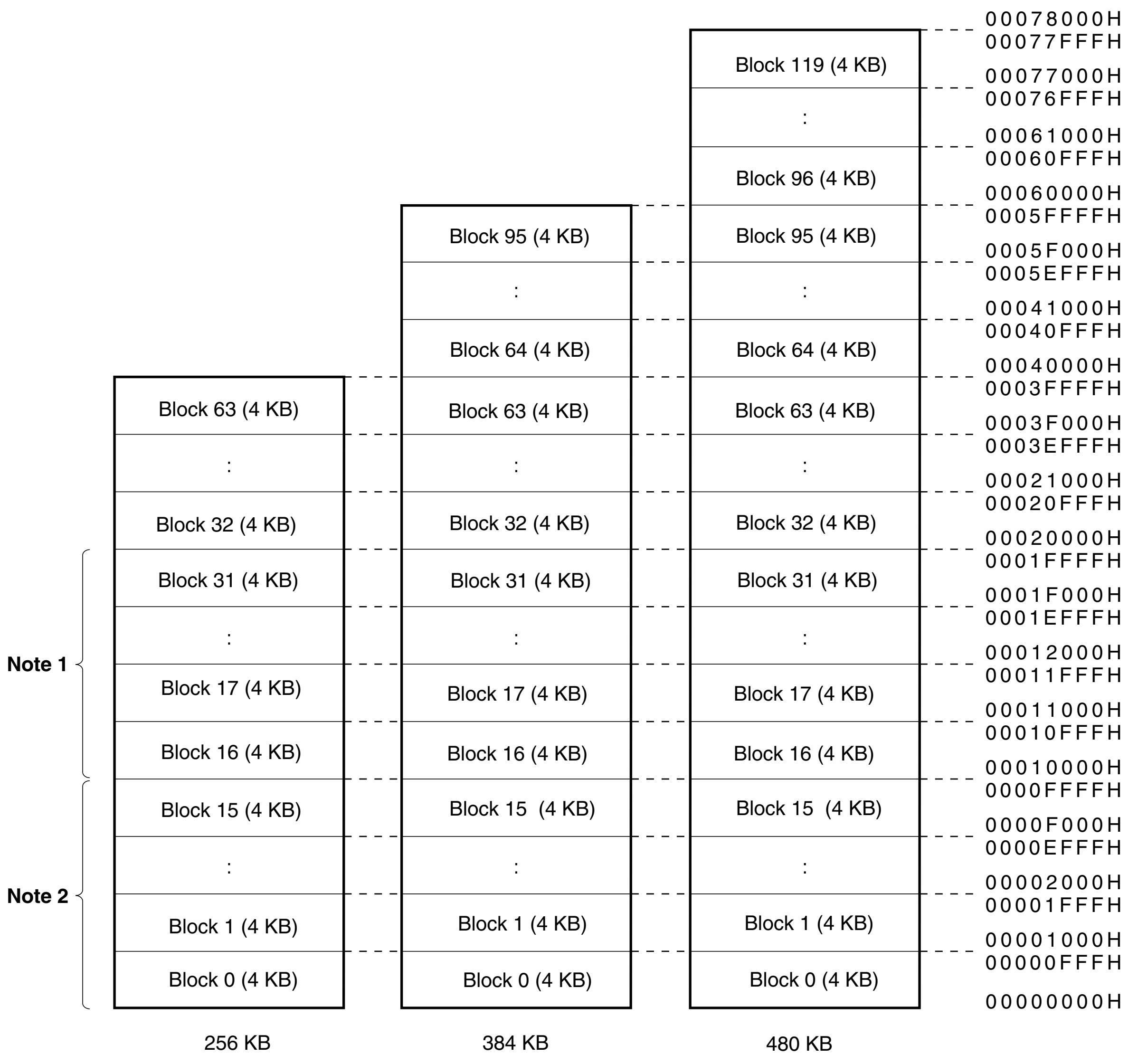
○ Rewriting flash memory and read disable function supported (security enforced)
○ Interrupts can be acknowledged during self programming.

27.2 Memory Configuration

The internal flash memory area of the V850E/IG4-H and V850E/IH4-H is divided into 64, 96, or 120 blocks and can be programmed/erased in block units. All the blocks can also be erased at once.

When the boot swap function is used, the physical memory located at the addresses of blocks 0 to 15 is replaced by the physical memory located at the addresses of blocks 16 to 31. For details of the boot swap function, see **27.9 Rewriting by Self Programming**.

Figure 27-1. Flash Memory Mapping



Notes 1. Area to be replaced with the boot area by the boot swap function

2. Boot area

27.3 Functional Overview

The internal flash memory of the V850E/IG4-H and V850E/IH4-H can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the V850E/IG4-H and V850E/IH4-H have already been mounted on the target system or not (off-board/on-board programming).

In addition, a security function that prohibits rewriting the user program written to the internal flash memory is also supported, so that the program cannot be changed by an unauthorized person.

The rewrite function using the user program (self programming) is ideal for an application where it is assumed that the program is changed after production/shipment of the target system. A boot swap function that rewrites the entire flash memory area safely is also supported. In addition, interrupt servicing is supported during self programming, so that the flash memory can be rewritten under various conditions, such as while communicating with an external device.

Table 27-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash memory programmer.	Flash memory programming mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash memory programmer and a dedicated program adapter board (FA series).	
Self programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of on-board/off-board programming. (During self programming, instructions cannot be fetched from or data access cannot be made to the on-chip flash memory area. Therefore, the rewrite program must be transferred to the internal RAM in advance).	Normal operation mode

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 27-2. Basic Functions

Function	Functional Outline	Support (√: Supported, ×: Not supported)	
		On-Board/Off-Board Programming	Self Programming
Block erasure	The contents of specified memory blocks are erased.	√	√
Chip erasure	The contents of the entire memory area are erased all at once.	√	× (supported by specifying area for block erasure)
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	√	√
Verify/checksum	Data read from the flash memory is compared with data transferred from the flash memory programmer.	√	× (Can be read by user program)
Blank check	The erasure status of the entire memory is checked.	√	√
Security setting	Use of the block erase command, chip erase command, program command, and read command can be prohibited.	√	× (Only values set by on-board/off-board programming can be retained)

Table 27-3. Security Functions

Function	Function Outline	Support	
		On-Board/Off-Board Programming	Self Programming
Block erase command prohibit	Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.	For details, see **27.3.2 Security function**.	
Chip erase command prohibit	Execution of block erase and chip erase commands on all blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.		
Program command prohibit	Write and block erase commands on all blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.		
Read command prohibit	Read command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.		

27.3.1 Erase units

(1) All area batch erase

Flash memory area 256 KB, 384 KB, or 480 KB can be erased at the same time.

(2) Erase in block units

Can be erased in block units.

- μPD70F3919 (V850E/IG4-H), 70F3922 (V850E/IH4-H): Block 0 to block 63: Each 4 KB
- μPD70F3920 (V850E/IG4-H), 70F3923 (V850E/IH4-H): Block 0 to block 95: Each 4 KB
- μPD70F3921 (V850E/IG4-H), 70F3924 (V850E/IH4-H): Block 0 to block 119: Each 4 KB

27.3.2 Security function

The commands and functions can be secured when the flash memory is rewritten.

As a factory-set condition in the V850E/IG4-H and V850E/IH4-H, “All enabled” is selected and the flash memory to which nothing has been written is secured.

Table 27-4. Security Setting

Function	Erase, Write, Read Operations When Each Security Is Set (√: Executable, ×: Not Executable, –: Not Supported)		Notes on Security Setting	
	On-Board/ Off-Board Programming	Self Programming	On-Board/ Off-Board Programming	Self Programming
Block erase command prohibit	Block erase command: × Chip erase command: √ Program command: √ Read command: √	Block erasure: √ Chip erasure: – Write: √	Setting of prohibition can be initialized by chip erase command.	Supported only when setting is changed from enable to prohibit
Chip erase command prohibit	Block erase command: × Chip erase command: × Program command: √[Note 1] Read command: √	Block erasure: √ Chip erasure: – Write: √	Setting of prohibition cannot be initialized.	
Program command prohibit	Block erase command: × Chip erase command: √ Program command: × Read command: √	Block erasure: √ Chip erasure: – Write: √	Setting of prohibition can be initialized by chip erase command.	
Read command prohibit	Block erase command: √ Chip erase command: √ Program command: √ Read command: ×	Block erasure: √ Chip erasure: – Write: √		
Boot area rewrite prohibit	Block erase command: ×[Note 2] Chip erase command: × Program command: ×[Note 2] Read command: √	Block erasure: √ Chip erasure: – Write: √	Setting of prohibition cannot be initialized.	

Notes 1. In this case, since the erase command is invalid, data different from the data already written in the flash memory cannot be written.

2. Executable except in boot area.

27.4 Writing with Flash Memory Programmer

Writing can be performed either on-board or off-board using a flash memory programmer (PG-FP4, PG-FP5, FL-PR4, or FL-PR5) and MINICUBE2.

(1) On-board programming

The contents of the flash memory are rewritten after the V850E/IG4-H or V850E/IH4-H is mounted on the target system. Mount connectors, etc., on the target system to connect the flash memory programmer.

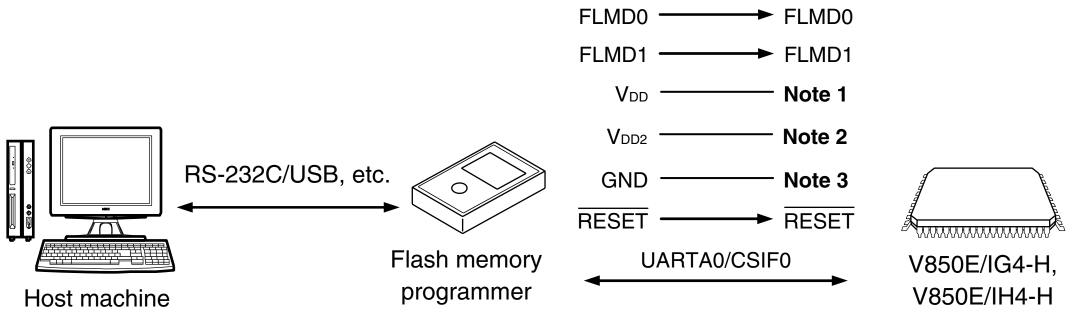
(2) Off-board programming

Writing to a flash memory is performed before mounting the V850E/IG4-H or V850E/IH4-H on the target system.

Remark FL-PR4 and FL-PR5 are products of Naito Densei Machida Mfg. Co., Ltd.

27.5 Flash Memory Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850E/IG4-H and V850E/IH4-H.



Notes 1. EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3} (V850E/IH4-H only), FV_{DD} (V850E/IH4-H only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{REFP0}, AV_{REFP1}

2. V_{DD0}, V_{DD1}, V_{DD2}

3. V_{SS0}, V_{SS1}, V_{SS2}, EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), EV_{SS4}, AV_{SS0}, AV_{SS1}, AV_{SS2}

A host machine is required for controlling the flash memory programmer.

UARTA0 or CSIF0 is used for the interface between the flash memory programmer and the V850E/IG4-H or V850E/IH4-H to perform writing, erasing, etc. Supply the operating clock of the V850E/IG4-H or V850E/IH4-H via the oscillator configured on the V850E/IG4-H or V850E/IH4-H board using a resonator and a capacitor.

Table 27-5. Environment and Communication Mode

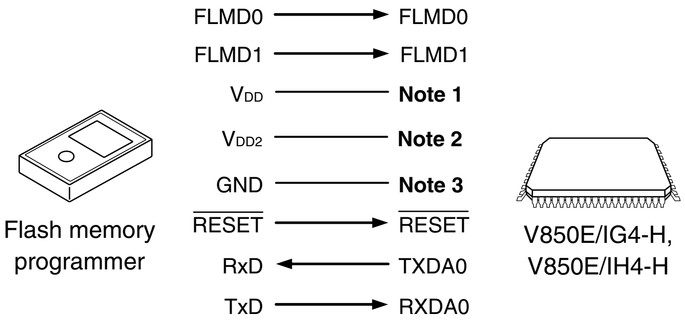
Environment	Communication Mode		
	UARTA0	CSIF0	CSIF0 for Handshake
Flash memory programmer (PG-FP4, PG-FP5, FL-PR4, and FL-PR5)	√	√	√
MINICUBE2	√	×	√

Remark √: Supported, ×: Not supported

27.6 Communication Method of Flash Memory Programming

(1) UART0 communication method

Transfer rate: 9,600 to 153,600 bps (LSB first)



Notes 1. EV$_{DD0}$, EV$_{DD1}$, EV$_{DD2}$, EV$_{DD3}$ (V850E/IH4-H only), FV$_{DD}$ (V850E/IH4-H only), AV$_{DD0}$, AV$_{DD1}$, AV$_{DD2}$, AV$_{REFP0}$, AV$_{REFP1}$

2. V$_{DD0}$, V$_{DD1}$, V$_{DD2}$

3. V$_{SS0}$, V$_{SS1}$, V$_{SS2}$, EV$_{SS0}$, EV$_{SS1}$, EV$_{SS2}$, EV$_{SS3}$ (V850E/IH4-H only), EV$_{SS4}$, AV$_{SS0}$, AV$_{SS1}$, AV$_{SS2}$

Cautions 1. Supply the operating clock of the V850E/IG4-H or V850E/IH4-H via the oscillator set up on the V850E/IG4-H or V850E/IH4-H board using a resonator and a capacitor.

2. For details, refer to the user's manual of each programmer.

Table 27-6. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IG4-H, V850E/IH4-H (1/2)

Pin No.	Dedicated Flash Memory Programmer (PG-FP4 or PG-FP5)	I/O (PG-FP4 or PG-FP5 Side)	V850E/IG4-H, V850E/IH4-H		
			Pin Name	Pin No.	
				IG4-H	IH4-H
				GC	GF
1	GND	–	V_{SS0}	38	81
			V_{SS1}	64	117
			V_{SS2}	91	28
			EV_{SS0}	41	85
			EV_{SS1}	63	116
			EV_{SS2}	100	38
			EV_{SS3}[Note 1]	–	8
			EV_{SS4}	31	74
			AV_{SS0}	5	43
			AV_{SS1}	10	48
			AV_{SS2}	27	66
2	$\overline{\text{RESET}}$	Output	$\overline{\text{RESET}}$	39	82
3	SI/RxD	Input	TXDA0	47	97
4	V_{DD}	–	EV_{DD0}	40	83
			EV_{DD1}	62	115
			EV_{DD2}	99	37
			EV_{DD3}[Note 1]	–	7
			FV_{DD}[Note 1]	–	114
			AV_{DD0}	7	45
			AV_{DD1}	8	46
			AV_{DD2}	26	65
			AV_{REFP0}	6	44
			AV_{REFP1}	9	47
5	SO/TxD	Output	RXDA0	46	96
6	V_{PP}	×	NC	–	–
7	SCK	×	NC	–	–
8	H/S	×	NC	–	–
9	CLK[Note 2]	Output	X1[Note 2]	36	79
10	VDE	×	NC	–	–

Notes 1. V850E/IH4-H only

2. In the V850E/IG4-H and V850E/IH4-H, external clock input is prohibited. Mount the resonator on the board.

Remark NC: No Connection
IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

Table 27-6. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IG4-H, V850E/IH4-H (2/2)

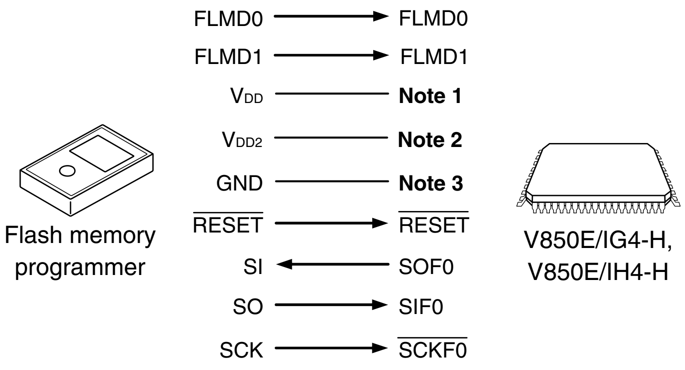
Pin No.	Dedicated Flash Memory Programmer (PG-FP4 or PG-FP5)	I/O (PG-FP4 or PG-FP5 Side)	V850E/IG4-H, V850E/IH4-H		
			Pin Name	Pin No.	
				IG4-H	IH4-H
				GC	GF
11	V_{DD2}	–	V_{DD0}	35	78
			V_{DD1}	65	118
			V_{DD2}	90	27
12	FLMD1	Output	**Note**	76	1
13	RFU-1	×	NC	–	–
14	FLMD0	Output	FLMD0	42	86
15	Not used	×	NC	–	–
16	Not used	×	NC	–	–

Note Connect to FLMD1 or GND by way of a resistor.

Remark NC: No Connection
IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(2) CSIF0 communication method

Serial clock: 5 MHz or less (MSB first)



Notes 1. EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3} (V850E/IH4-H only), FV_{DD} (V850E/IH4-H only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{REFP0}, AV_{REFP1}

2. V_{DD0}, V_{DD1}, V_{DD2}

3. V_{SS0}, V_{SS1}, V_{SS2}, EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), EV_{SS4}, AV_{SS0}, AV_{SS1}, AV_{SS2}

Cautions 1. Supply the operating clock of the V850E/IG4-H or V850E/IH4-H via the oscillator configured on the V850E/IG4-H or V850E/IH4-H board using a resonator and a capacitor.

2. For details, refer to the user's manual of each programmer.

The flash memory programmer outputs (master) transfer clocks and the V850E/IG4-H or V850E/IH4-H operates as a slave.

Table 27-7. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IG4-H, V850E/IH4-H (1/2)

Pin No.	Dedicated Flash Memory Programmer (PG-FP4 or PG-FP5)	I/O (PG-FP4 or PG-FP5 Side)	V850E/IG4-H, V850E/IH4-H		
			Pin Name	Pin No.	
				IG4-H	IH4-H
				GC	GF
1	GND	–	V$_{SS0}$	38	81
			V$_{SS1}$	64	117
			V$_{SS2}$	91	28
			EV$_{SS0}$	41	85
			EV$_{SS1}$	63	116
			EV$_{SS2}$	100	38
			EV$_{SS3}$[Note 1]	–	8
			EV$_{SS4}$	31	74
			AV$_{SS0}$	5	43
			AV$_{SS1}$	10	48
			AV$_{SS2}$	27	66
2	$\overline{\text{RESET}}$	Output	$\overline{\text{RESET}}$	39	82
3	SI/RxD	Input	SOF0	47	97
4	V$_{DD}$	–	EV$_{DD0}$	40	83
			EV$_{DD1}$	62	115
			EV$_{DD2}$	99	37
			EV$_{DD3}$[Note 1]	–	7
			FV$_{DD}$[Note 1]	–	114
			AV$_{DD0}$	7	45
			AV$_{DD1}$	8	46
			AV$_{DD2}$	26	65
			AV$_{REFP0}$	6	44
			AV$_{REFP1}$	9	47
5	SO/TxD	Output	SIF0	46	96
6	V$_{PP}$	×	NC	–	–
7	SCK	Output	$\overline{\text{SCKF0}}$	48	98
8	H/S	×	NC	–	–
9	CLK[Note 2]	Output	X1[Note 2]	36	79
10	VDE	×	NC	–	–

Notes 1. V850E/IH4-H only

2. In the V850E/IG4-H and V850E/IH4-H, external clock input is prohibited. Mount the resonator on board.

Remark NC: No Connection
IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

Table 27-7. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IG4-H, V850E/IH4-H (2/2)

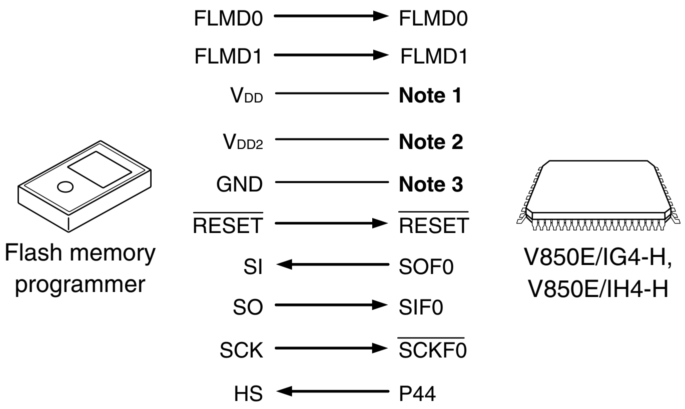
Pin No.	Dedicated Flash Memory Programmer (PG-FP4 or PG-FP5)	I/O (PG-FP4 or PG-FP5 Side)	V850E/IG4-H, V850E/IH4-H		
			Pin Name	Pin No.	
				IG4-H	IH4-H
				GC	GF
11	V_{DD2}	–	V_{DD0}	35	78
			V_{DD1}	65	118
			V_{DD2}	90	27
12	FLMD1	Output	**Note**	76	1
13	RFU-1	×	NC	–	–
14	FLMD0	Output	FLMD0	42	86
15	Not used	×	NC	–	–
16	Not used	×	NC	–	–

Note Connect to FLMD1 or GND by way of a resistor.

Remark NC: No Connection
IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

(3) CSIF0 communication method supporting handshake

Serial clock: 5 MHz or less (MSB first)



Notes 1. EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3} (V850E/IH4-H only), FV_{DD} (V850E/IH4-H only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{REFP0}, AV_{REFP1}

2. V_{DD0}, V_{DD1}, V_{DD2}

3. V_{SS0}, V_{SS1}, V_{SS2}, EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), EV_{SS4}, AV_{SS0}, AV_{SS1}, AV_{SS2}

Cautions 1. Supply the operating clock of the V850E/IG4-H or V850E/IH4-H via the oscillator configured on the V850E/IG4-H or V850E/IH4-H board using a resonator and a capacitor.

2. For details, refer to the user's manual of each programmer.

The flash memory programmer outputs the transfer clock, and the V850E/IG4-H or V850E/IH4-H operates as a slave.

When the PG-FP4 or PG-FP5 is used, it sends the following signals to the V850E/IG4-H or V850E/IH4-H. For details, refer to the **PG-FP4 User's Manual (U15260E)** or **PG-FP5 User's Manual (U18865E)**.

Table 27-8. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IG4-H, V850E/IH4-H (1/2)

Pin No.	Dedicated Flash Memory Programmer (PG-FP4 or PG-FP5)	I/O (PG-FP4 or PG-FP5 Side)	V850E/IG4-H, V850E/IH4-H		
			Pin Name	Pin No.	
				IG4-H	IH4-H
				GC	GF
1	GND	–	V_{SS0}	38	81
			V_{SS1}	64	117
			V_{SS2}	91	28
			EV_{SS0}	41	85
			EV_{SS1}	63	116
			EV_{SS2}	100	38
			EV_{SS3}[Note 1]	–	8
			EV_{SS4}	31	74
			AV_{SS0}	5	43
			AV_{SS1}	10	48
			AV_{SS2}	27	66
2	$\overline{\text{RESET}}$	Output	$\overline{\text{RESET}}$	39	82
3	SI/RxD	Input	SOF0	47	97
4	V_{DD}	–	EV_{DD0}	40	83
			EV_{DD1}	62	115
			EV_{DD2}	99	37
			EV_{DD3}[Note 1]	–	7
			FV_{DD}[Note 1]	–	114
			AV_{DD0}	7	45
			AV_{DD1}	8	46
			AV_{DD2}	26	65
			AV_{REFP0}	6	44
			AV_{REFP1}	9	47
5	SO/TxD	Output	SIF0	46	96
6	V_{PP}	×	NC	–	–
7	SCK	Output	$\overline{\text{SCKF0}}$	48	98
8	H/S	Input	P44	50	100
9	CLK[Note 2]	Output	X1[Note 2]	36	79
10	VDE	×	NC	–	–

Notes 1. V850E/IH4-H only

2. In the V850E/IG4-H and V850E/IH4-H, external clock input is prohibited. Mount the resonator on board.

Remark NC: No Connection
IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

Table 27-8. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IG4-H, V850E/IH4-H (2/2)

Pin No.	Dedicated Flash Memory Programmer (PG-FP4 or PG-FP5)	I/O (PG-FP4 or PG-FP5 Side)	V850E/IG4-H, V850E/IH4-H		
			Pin Name	Pin No.	
				IG4-H	IH4-H
				GC	GF
11	V_{DD2}	–	V_{DD0}	35	78
			V_{DD1}	65	118
			V_{DD2}	90	27
12	FLMD1	Output	**Note**	76	1
13	RFU-1	×	NC	–	–
14	FLMD0	Output	FLMD0	42	86
15	Not used	×	NC	–	–
16	Not used	×	NC	–	–

Note Connect to FLMD1 or GND by way of a resistor.

Remark NC: No Connection
IG4-H: V850E/IG4-H
IH4-H: V850E/IH4-H
GC (V850E/IG4-H): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IH4-H): 128-pin plastic LQFP (fine pitch) (14 × 20)

27.7 Pin Processing During Flash Memory Programming

When performing on-board programming, mount a connector on the target system to connect to the flash memory programmer.

In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset in the normal operation mode. Therefore, because all the ports become high-impedance status, pin processing is required when the external device does not acknowledge the high-impedance status.

27.7.1 Power supply

Supply the same power supplies (V_{DD0}, V_{DD1}, V_{DD2}, V_{SS0}, V_{SS1}, V_{SS2}, EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3} (V850E/IH4-H only), EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), EV_{SS4}, FV_{DD} (V850E/IH4-H only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{SS0}, AV_{SS1}, AV_{SS2}, AV_{REFP0}, AV_{REFP1}) as in the normal operation mode. Connect V_{DD}, V_{DD2}, and GND of the flash memory programmer to V_{DD0}, V_{DD1}, V_{DD2}, V_{SS0}, V_{SS1}, V_{SS2}, EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3} (V850E/IH4-H only), EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), EV_{SS4}, FV_{DD} (V850E/IH4-H only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{SS0}, AV_{SS1}, AV_{SS2}, AV_{REFP0}, AV_{REFP1}. (V_{DD} of the flash memory programmer is provided with a power supply monitoring function.)

In the flash memory programming mode (including flash memory self programming), insert capacitors between V_{DD0}, V_{DD1}, V_{DD2} pins and V_{SS0}, V_{SS1}, V_{SS2} pins and EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3} (V850E/IH4-H only), FV_{DD} (V850E/IH4-H only) pins and EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), EV_{SS4} pins to stabilize the power supply voltage.

27.7.2 Pins used

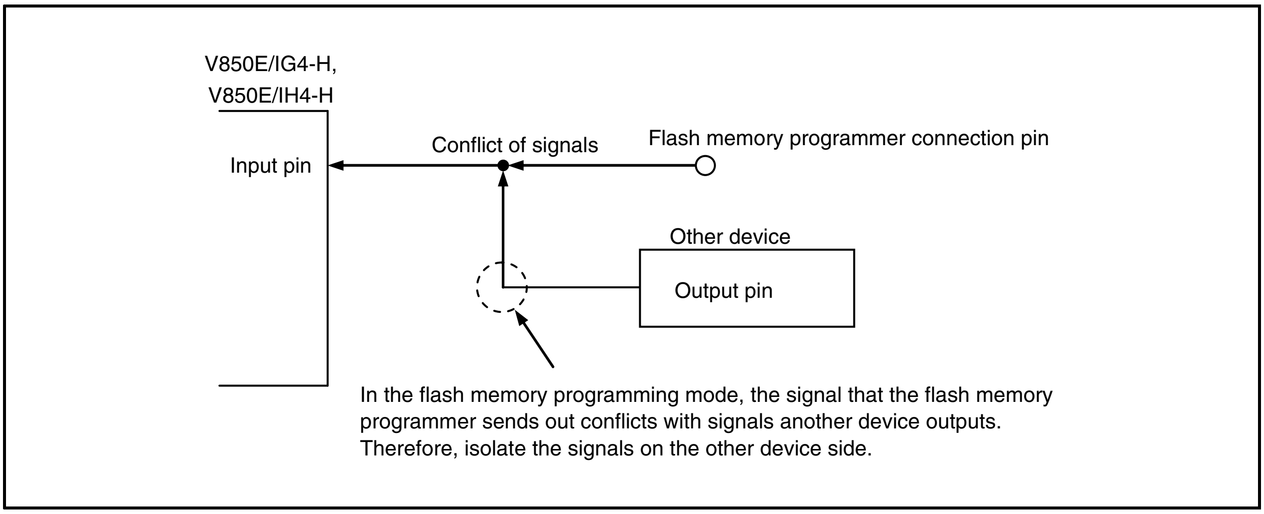
The following shows the pins used by each interface.

Communication Mode	Pins Used
UARTA0	TXDA0, RXDA0
CSIF0	SOF0, SIF0, $\overline{\text{SCKF0}}$
CSIF0 supporting handshake	SOF0, SIF0, $\overline{\text{SCKF0}}$, P44

When connecting a flash memory programmer to an interface pin that is connected to other devices on-board, care should be taken to avoid a conflict of signals or the malfunction of other devices.

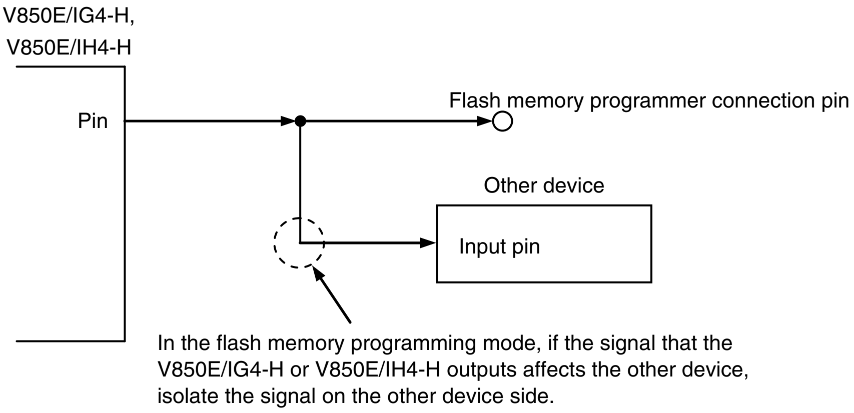
(1) Conflict of signals

When the flash memory programmer (output) is connected to an interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

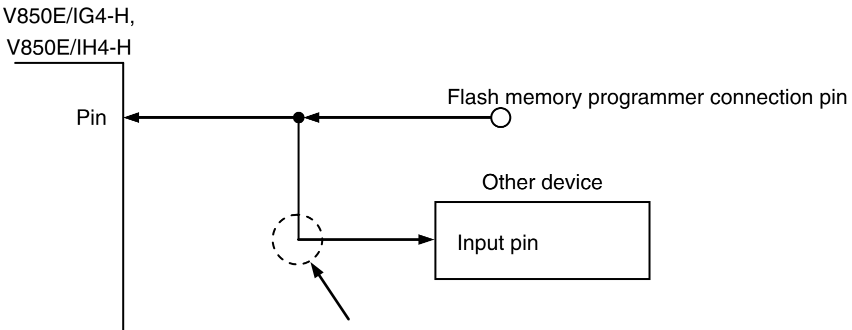

V850E/IG4-H,
V850E/IH4-H
Input pin
Conflict of signals
Flash memory programmer connection pin
Other device
Output pin
In the flash memory programming mode, the signal that the flash memory programmer sends out conflicts with signals another device outputs. Therefore, isolate the signals on the other device side.

(2) Malfunction of other device

When the flash memory programmer (output or input) is connected to an interface pin (input or output) that is connected to another device (input), the signal is output to the other device, causing the device to malfunction. To avoid this, isolate the connection to the other device or make the setting so that the input signal to the other device is ignored.



In the flash memory programming mode, if the signal that the V850E/IG4-H or V850E/IH4-H outputs affects the other device, isolate the signal on the other device side.

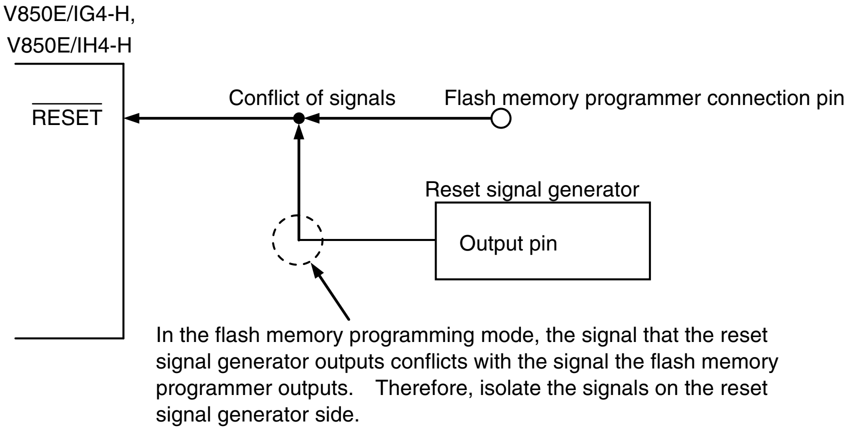


In the flash memory programming mode, if the signal that the flash memory programmer outputs affects the other device, isolate the signal on the other device side.

27.7.3 $\overline{\text{RESET}}$ pin

When the reset signal of the flash memory programmer is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

When a reset signal is input from the user system in the flash memory programming mode, the programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the flash memory programmer.

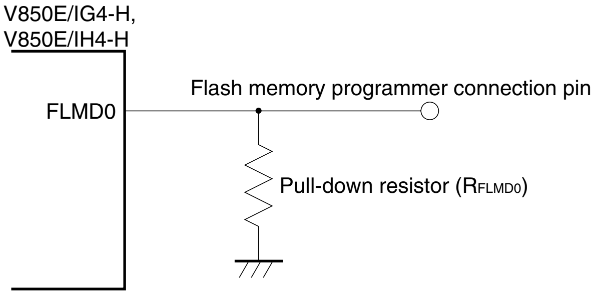

V850E/IG4-H,
V850E/IH4-H
$\overline{\text{RESET}}$
Conflict of signals
Flash memory programmer connection pin
Reset signal generator
Output pin
In the flash memory programming mode, the signal that the reset signal generator outputs conflicts with the signal the flash memory programmer outputs. Therefore, isolate the signals on the reset signal generator side.

27.7.4 FLMD0 and FLMD1 pins

(1) FLMD0 pin

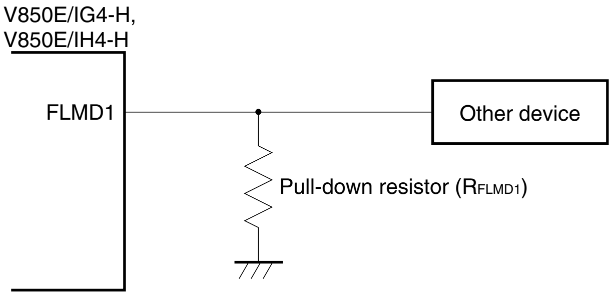
In the normal operation mode, input a voltage of EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), or EV_{SS4} level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of EV_{DD0}, EV_{DD1}, EV_{DD2}, or EV_{DD3} (V850E/IH4-H only) level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode, a voltage of EV_{DD0}, EV_{DD1}, EV_{DD2}, or EV_{DD3} (V850E/IH4-H only) level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory. For details, see **27.9.5 (1) FLMD0 pin**.


V850E/IG4-H,
V850E/IH4-H
FLMD0
Flash memory programmer connection pin
Pull-down resistor (R_{FLMD0})

(2) FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When EV_{DD0}, EV_{DD1}, EV_{DD2}, or EV_{DD3} (V850E/IH4-H only) is supplied to the FLMD0 pin, the flash memory programming mode is entered, so 0 V must be input to the FLMD1 pin. The following shows an example of the connection of the FLMD1 pin.



Caution If the EV_{DD0}, EV_{DD1}, EV_{DD2}, or EV_{DD3} (V850E/IH4-H only) signal is input to the FLMD1 pin from another device during on-board programming and immediately after reset, isolate this signal.

Table 27-9. Relationship Between FLMD0 and FLMD1 Pins and Operation Mode When Reset Ends

FLMD0	FLMD1	Operation Mode
0	Either	Normal operation mode
EV_{DD}	0	Flash memory programming mode
EV_{DD}	EV_{DD}	Setting prohibited

Remark EV_{DD}: EV_{DD0}, EV_{DD1}, EV_{DD2}, and EV_{DD3} (V850E/IH4-H only)

27.7.5 Port pins

When the flash memory programming mode is set, all the port pins except the pin that communicates with the flash memory programmer change to the high-impedance status. These port pins need not be processed. If problems such as disabling of the high-impedance status should occur to the external devices connected to the ports, connect them to EV_{DD0}, EV_{DD1}, EV_{DD2}, and EV_{DD3} (V850E/IH4-H only), or EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), and EV_{SS4} by way of resistors.

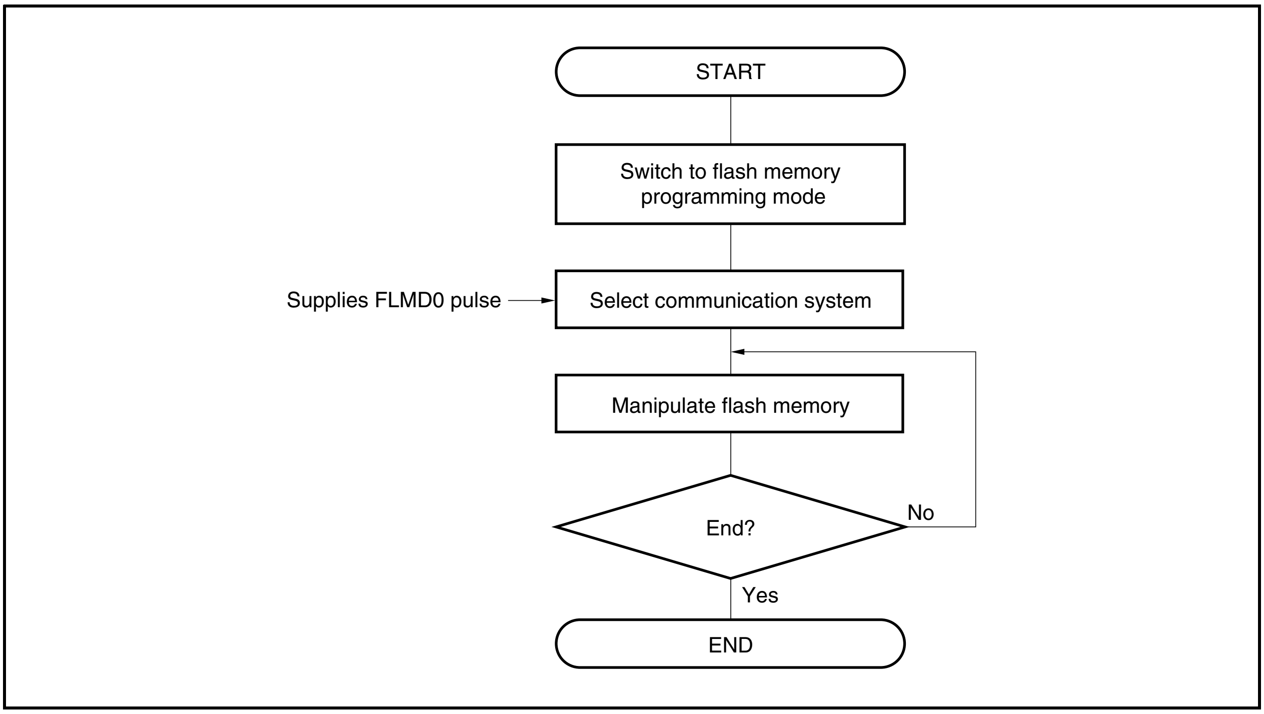
27.7.6 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode.

27.8 Flash Memory Programming Mode

27.8.1 Flash memory control

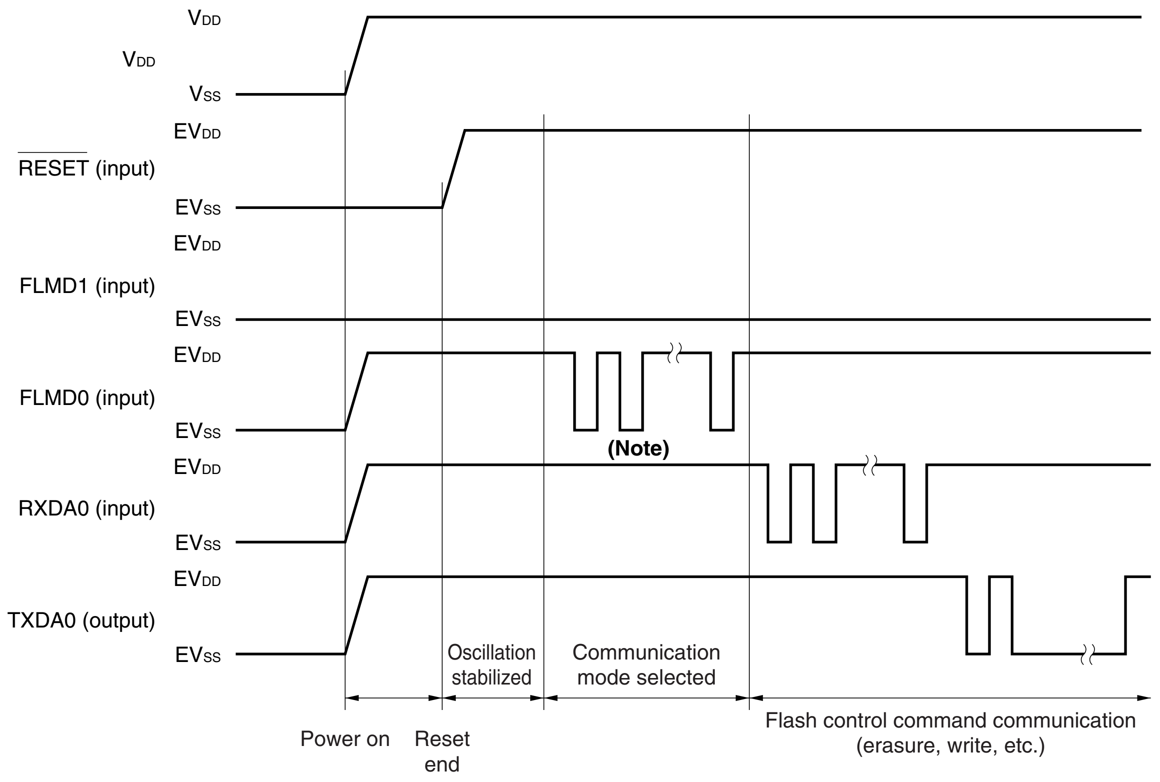
The following shows the procedure for manipulating the flash memory.


START
Switch to flash memory programming mode
Supplies FLMD0 pulse
Select communication system
Manipulate flash memory
End?
No
Yes
END

27.8.2 Selection of communication mode

In the V850E/IG4-H and V850E/IH4-H, the communication mode is selected by inputting pulses (11 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the flash memory programmer.

The following shows the relationship between the number of pulses and the communication mode.



Note The number of clocks is as follows depending on the communication mode.

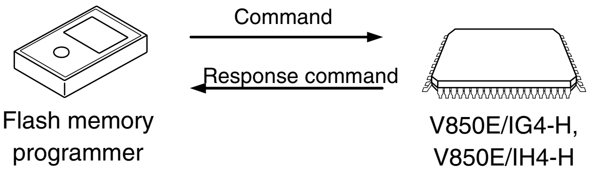
FLMD0 pulse	Communication mode	Remarks
0	UARTA0	Communication rate: 9,600 bps (after reset), LSB first
8	CSIF0	V850E/IG4-H and V850E/IH4-H perform slave operation, MSB first
11	CSIF0 for handshake	V850E/IG4-H and V850E/IH4-H perform slave operation, MSB first
Other	RFU	Setting prohibited

Caution When UARTA0 is selected, the receive clock is calculated based on the reset command sent from the flash memory programmer after receiving the FLMD0 pulse.

Remark V_{DD}: V_{DD0}, V_{DD1}, V_{DD2}
EV_{DD}: EV_{DD0}, EV_{DD1}, EV_{DD2}, and EV_{DD3} (V850E/IH4-H only)
EV_{SS}: EV_{SS0}, EV_{SS1}, EV_{SS2}, EV_{SS3} (V850E/IH4-H only), and EV_{SS4}

27.8.3 Communication commands

The V850E/IG4-H and V850E/IH4-H communicate with a flash memory programmer by means of commands. The commands sent from the flash memory programmer to the V850E/IG4-H or V850E/IH4-H are called "commands". The response signals sent from the V850E/IG4-H or V850E/IH4-H to the flash memory programmer are called "response commands".



The following shows the commands for flash memory control in the V850E/IG4-H and V850E/IH4-H. All of these commands are issued from the dedicated flash memory programmer, and the V850E/IG4-H and V850E/IH4-H perform the processing corresponding to the commands.

Table 27-10. Flash Memory Control Commands

Classification	Command Name	Support			Function
		UARTA0	CSIF0	**Note**	
Blank check	Block blank check command	√	√	√	Checks if the contents of the memory in the specified block have been correctly erased.
Erase	Chip erase command	√	√	√	Erases the contents of the entire memory.
	Block erase command	√	√	√	Erases the contents of the memory of the specified block.
Write	Program command	√	√	√	Writes the specified address range, and executes a contents verify check.
Verify	Verify command	√	√	√	Compares the contents of memory in the specified address range with data transferred from the flash memory programmer.
	Checksum command	√	√	√	Reads the checksum in the specified address range.
System setting and control	Silicon signature command	√	√	√	Reads silicon signature information.
	Security setting command	√	√	√	Prohibits the chip erase command, block erase command, program command, read command, and boot area rewrite.

Note CSIF0 supporting handshake

The V850E/IG4-H and V850E/IH4-H send back response commands for the commands issued from the flash memory programmer. The response commands sent from the V850E/IG4-H and V850E/IH4-H are listed below.

Table 27-11. Response Commands

Response Command Name	Function
ACK (Acknowledge)	Acknowledges command/data, etc.
NAK (Not acknowledge)	Acknowledges illegal frame, etc.
Command number error	Acknowledges illegal command/data, etc.
Parameter error	Acknowledges illegal parameter, etc.
Checksum error	Acknowledges checksum of frame
Protect error	Acknowledges when protection is in effect
During processing (BUSY)	Acknowledges during processing
Other than above	Error

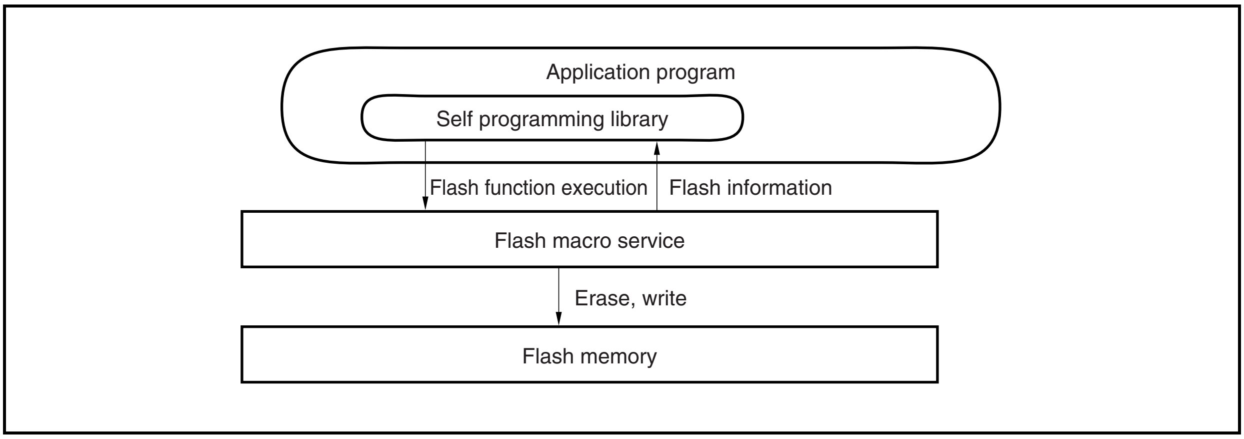
27.9 Rewriting by Self Programming

27.9.1 Overview

The V850E/IG4-H and V850E/IH4-H support a flash macro service that allows the user program to rewrite the internal flash memory by itself. By using this interface and a self programming library that is used to rewrite the flash memory with a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM or external memory. Consequently, the user program can be upgraded and constant data[Note] can be rewritten in the field. For details about self programming, see **Flash Memory Self Programming Library User's Manual**.

Note Be sure not to allocate the program code to the block where the constant data of rewriting target is allocated. See **27.2 Memory Configuration** for the block configuration.

Figure 27-2. Concept of Self Programming



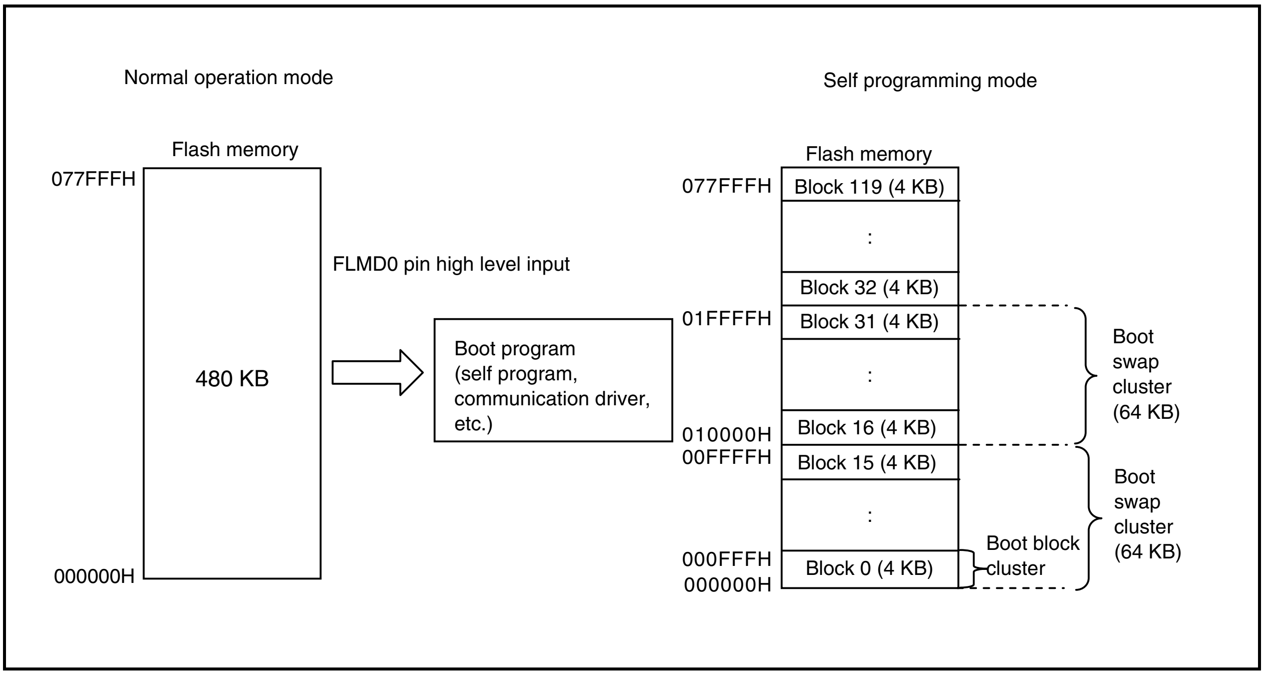
27.9.2 Features

(1) Flash memory self programming

Flash memory self programming is used to erase or write the flash memory by calling the flash function from a program stored in an area other than the flash memory area to be erased or written. To store the program that implements self programming in the area to be erased or written, copy the program to the internal RAM area, execute the program at the copy destination, and call the flash function.

To call the flash function, change the mode from the normal operation mode to the self programming mode by using the flash programming mode control register.

Figure 27-3. Self Programming



(a) Boot swap cluster

The contents of the boot swap cluster of the lower address side (000000H to 00FFFFH) and the boot swap cluster of the higher address side (010000H to 01FFFFH) can be interchanged while flash memory programming is performed.

(b) Boot block cluster

By specifying the boot block cluster from 000000H in 4 KB units, the contents of the boot block cluster can be protected from rewriting.

(2) Interrupt support

Instructions cannot be fetched from the flash memory during self-programming. Consequently, a user handler written to the flash memory could not be used even if an interrupt has occurred.

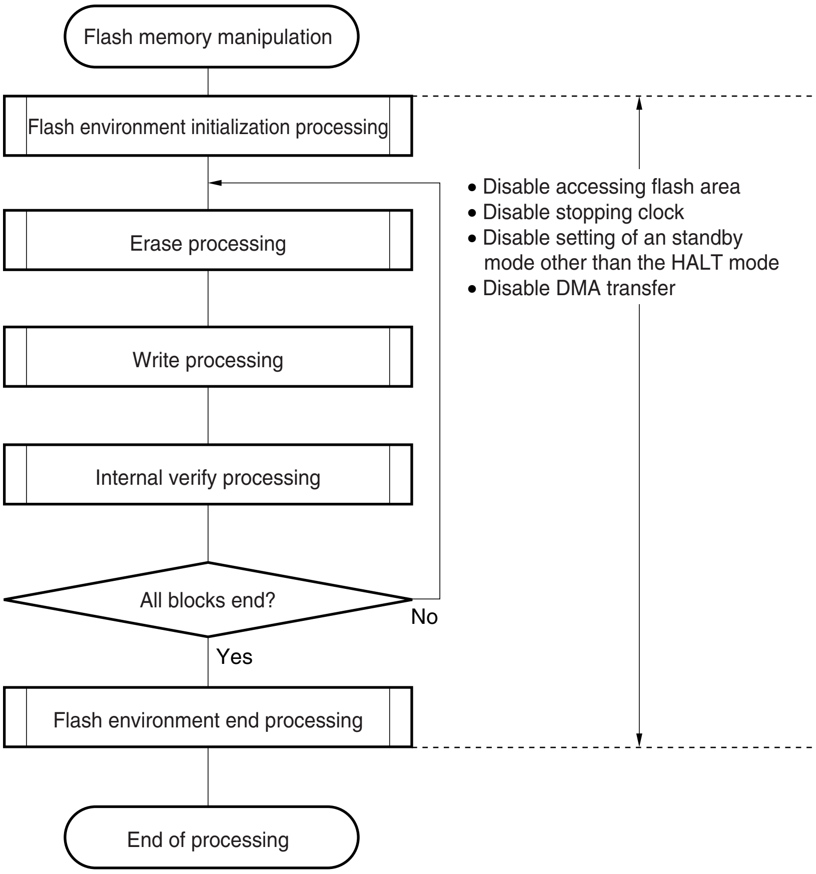
Therefore, in the V850E/IG4-H and V850E/IH4-H, to use an interrupt during self-programming, processing transits to the specific address[Note] in the internal RAM. Allocate the jump instruction that transits processing to the user interrupt servicing at the specific address[Note] in the internal RAM.

Note	NMI interrupt:	Start address of internal RAM
	Maskable interrupt:	Start address of internal RAM + 4 addresses

27.9.3 Standard self programming flow

The entire processing to rewrite the flash memory by flash self programming is illustrated below.

Figure 27-4. Standard Self Programming Flow


Flash memory manipulation
Flash environment initialization processing
• Disable accessing flash area
• Disable stopping clock
• Disable setting of an standby mode other than the HALT mode
• Disable DMA transfer
Erase processing
Write processing
Internal verify processing
All blocks end?
No
Yes
Flash environment end processing
End of processing

27.9.4 Flash functions

Table 27-12. Flash Function List

Function Name	Outline	Support
FlashInit	Self-programming library initialization	√
FlashEnv	Flash environment start/end	√
FlashFLMDCheck	FLMD pin check	√
FlashStatusCheck	Hardware processing execution status check	√
FlashBlockErase	Block erase	√
FlashWordWrite	Data write	√
FlashBlockIVerify	Internal verification of block	√
FlashBlockBlankCheck	Blank check of block	√
FlashSetInfo	Flash information setting	√
FlashGetInfo	Flash information acquisition	√
FlashBootSwap	Boot swap execution	√

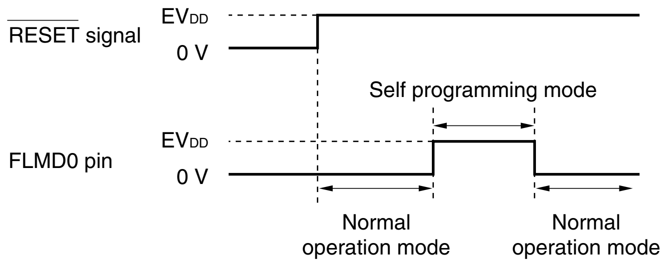
27.9.5 Pin processing

(1) FLMD0 pin

The FLMD0 pin is used to set the operation mode when reset ends and to protect the flash memory from being written during self rewriting. It is therefore necessary to keep the voltage applied to the FLMD0 pin at 0 V when reset ends and a normal operation is executed. It is also necessary to apply a voltage of EV_{DD0}, EV_{DD1}, EV_{DD2}, and EV_{DD3} (V850E/IH4-H only) level to the FLMD0 pin during the self programming mode period via port control before the memory is rewritten.

When self programming has been completed, the voltage on the FLMD0 pin must be returned to 0 V.

Figure 27-5. Mode Change Timing



Remark EV_{DD}: EV_{DD0}, EV_{DD1}, EV_{DD2}, and EV_{DD3} (V850E/IH4-H only)

Caution Make sure that the FLMD0 pin is at 0 V when reset ends.

27.9.6 Internal resources used

The following table lists the internal resources used for self programming. These internal resources can also be used freely for purposes other than self programming.

Table 27-13. Internal Resources Used

Resource Name	Description
Stack area	An extension of the stack used by the user is used by the library (can be used in both the internal RAM and external RAM).
Flash macro service area[Note]	A 9 KB internal RAM area (3FFCC00H to 3FFEFFFH)
Library code[Note]	Program entity of library (can be used anywhere other than the flash memory block to be manipulated).
Application program	Executed as user application. Calls flash functions.
Maskable interrupt	Can be used in the user application execution status or self-programming status. To use this interrupt in the self-programming status, since the processing transits to the address of the internal RAM start address + 4 addresses, allocate the jump instruction that transits the processing to the user interrupt servicing at the address of the internal RAM start address + 4 addresses in advance.
NMI interrupt	Can be used in the user application execution status or self-programming status. To use this interrupt in the self-programming status, since the processing transits to the address of the internal RAM start address, allocate the jump instruction that transits the processing to the user interrupt servicing at the internal RAM start address in advance.

Note About resources used, refer to the **Flash Memory Self-Programming Library User’s Manual**.

CHAPTER 28 ELECTRICAL SPECIFICATIONS

28.1 V850E/IG4-H

28.1.1 Absolute maximum ratings

(T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.5 to +2.0	V
	V_{SS}	V_{SSn} = EV_{SSm} = AV_{SSn}		–0.5 to +0.5	V
	EV_{DD}			–0.5 to +6.5	V
	EV_{SS}	V_{SSn} = EV_{SSm} = AV_{SSn}		–0.5 to +0.5	V
	AV_{DD}			–0.5 to +6.5	V
	AV_{SS}	V_{SSn} = EV_{SSm} = AV_{SSn}		–0.5 to +0.5	V
	UV_{DD}			–0.5 to +4.6	V
Input voltage	V_{I1}	**Note 1**		–0.5 to EV_{DD} + 0.5$^{\text{Note 2}}$	V
	V_{I2}	X1, X2		–0.5 to V_{DD} + 0.35	V
Output current, low	I_{OL}	All pins	Per pin	4	mA
			Total of all pins	63	mA
Output current, high	I_{OH}	All pins	Per pin	–4	mA
			Total of all pins	–63	mA
Analog input voltage	V_{IAN}	**Note 3**		–0.5 to AV_{DD} + 0.5$^{\text{Note 2}}$	V
Analog reference input voltage	V_{IREF}	AV_{REFP0}, AV_{REFP1}		–0.5 to AV_{DD} + 0.5$^{\text{Note 2}}$	V
Operating ambient temperature	T_A	In normal operating mode		–40 to +85	°C
		In flash memory programming mode		–40 to +85	°C
Storage temperature	T_{stg}			–40 to +125	°C

Notes 1. P00 to P07, P10 to P16, P20 to P24, P30 to P37, P40 to P44, P50 to P52, P70 to P711, PDL0 to PDL15, $\overline{\text{RESET}}$, FLMD0, $\overline{\text{DRST}}$

2. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

3. P70/ANI20 to P711/ANI211, ANI00/ANI05 to ANI02/ANI07, ANI03, ANI10/ANI15 to ANI12/ANI17

Cautions 1. Do not directly connect the output pins (or I/O pins in the output state) of IC products to other output pins (including I/O pins in the output state), power supply pins such as V_{DD} and EV_{DD}, or GND pin. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Remark n = 0 to 2

m = 0 to 2, 4

28.1.2 Capacitance

(T_A = 25°C, V_{DD0} = V_{SS0} = V_{DD1} = V_{SS1} = V_{DD2} = V_{SS2} = EV_{DD0} = EV_{SS0} = EV_{DD1} = EV_{SS1} = EV_{DD2} = EV_{SS2} = EV_{SS4} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = AV_{DD2} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	fc = 1 MHz Unmeasured pins returned to 0 V	**Note 1**			15	pF
I/O capacitance	C_{IO}		**Note 2**			15	pF
Output capacitance	C_O		**Note 3**			15	pF

Notes **1.** ANI00/ANI05 to ANI02/ANI07, ANI03, ANI10/ANI15 to ANI12/ANI17, $\overline{\text{RESET}}$

2. P00 to P07, P10 to P16, P24 to P27, P30 to P37, P40 to P44, P50 to P52, P70 to P711, PDL0 to PDL15

3. DDO

Cautions 1. Excludes the FLMD0, $\overline{\text{DRST}}$, X1, and X2 pins.

2. In addition to input capacitance, sampling capacitance is added to the ANI00/ANI05 to ANI02/ANI07, ANI03, ANI10/ANI15 to ANI12/ANI17, and ANI20 to ANI211 pins when sampling.

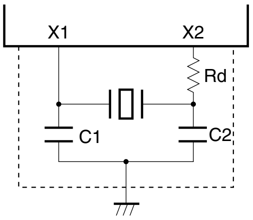
28.1.3 Operating conditions

(T_A = –40 to +85°C, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{XX}	PLL mode	80		100	MHz
		Clock through mode	10		12.5	MHz
CPU clock frequency	f_{CPU}	PLL mode	10		100	MHz
		Clock through mode	1.25		12.5	MHz
V_{DD} voltage	V_{DD}		1.35		1.65	V
EV_{DD} voltage	EV_{DD}		3.5		5.5	V
AV_{DD} voltage	AV_{DD}	When A/D converters 0 to 2 are operating	4.0		5.5	V
		When A/D converters 0 to 2 are not operating	3.5		5.5	V
UV_{DD} voltage	UV_{DD}		3.0		3.6	V

28.1.4 Clock oscillator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V, EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic /crystal resonator		Oscillation frequency (f_X)		10		12.5	MHz
		Oscillation stabilization time	After reset release		$2^{15}/f_X$		ms
			After STOP mode release		**Note**		ms

Note The value varies depending on the setting of the oscillation stabilization time select register (OSTS).

Cautions 1. Connect the oscillator as close to the X1 and X2 pins as possible.

2. Do not cross the wiring with the other signal lines in the area enclosed by the broken lines in the above figure.

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

4. Inputting an external clock to the V850E/IG4-H is prohibited.

28.1.5 DC characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V) (1/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**			$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	**Note 2**			$0.8EV_{DD}$		EV_{DD}	V
	V_{IH3}	**Note 3**			2.2		EV_{DD}	V
	V_{IH4}	**Note 4**			$0.7AV_{DD}$		AV_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**			EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	**Note 2**			EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	**Note 3**			EV_{SS}		0.8	V
	V_{IL4}	**Note 4**			AV_{SS}		$0.3AV_{DD}$	V
Input leakage current, high	I_{LIH1}	V_I = **Note 5,** **Note 6**	Other than X1				5	μA
	I_{LIH2}		X1				20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	Other than X1				–5	μA
	I_{LIL2}		X1				–20	μA
Output leakage current, high	I_{LOH}	V_O = **Note 5**					5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V					–5	μA
Output voltage, high	V_{OH1}	**Note 7**	I_{OH} = –1.0 mA	Total of pins = –52 mA	EV_{DD} – 1.0			V
Output voltage, low	V_{OL1}	**Note 7**	I_{OL} = 1.0 mA	Total of pins = 52 mA			0.4	V
Pull-up resistor	R_{L1}				10	30	120	kΩ
Pull-down resistor Note 8	R_{L2}				10	30	120	kΩ

Notes 1. P33, P36, P41, PDL0 to PDL15 pins

2. P00 to P07, P10 to P16, P24 to P27, P30 to P32, P34, P35, P37, P40, P42 to P44, P50 to P52, $\overline{\text{RESET}}$, FLMD0 pins

3. $\overline{\text{DRST}}$, DDI, DCK, and DMS pins

4. P70 to P711 pins

5. AV_{DD0} = AV_{DD1} = AV_{DD2} = EV_{DD0} = EV_{DD1} = EV_{DD2}

6. Except for $\overline{\text{DRST}}$ pin

7. P00 to P07, P10 to P16, P20 to P27, P30 to P37, P40 to P44, P50 to P52, PDL0 to PDL15, DDO pins

8. $\overline{\text{DRST}}$ pin only

Remark The characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.[Note 1]	MAX.	Unit
V_{DD} supply current[Note 2]	I_{DD1}	f_{XX} = 100 MHz	Normal operation		125	205	mA
	I_{DD2}		HALT mode		66	143	mA
	I_{DD3}		IDLE mode		6	50	mA
	I_{DD4}	STOP mode			0.1	16	mA

Notes 1. The TYP. value is a reference value when V_{DD0} = V_{DD1} = V_{DD2} = 1.5 V and T_A = 25°C.

2. The current consumed by the EV_{DD} system (output buffer and pull-up resistor) and the operating currents of A/D converters 0 to 2, the operational amplifier, and the comparator are not included.

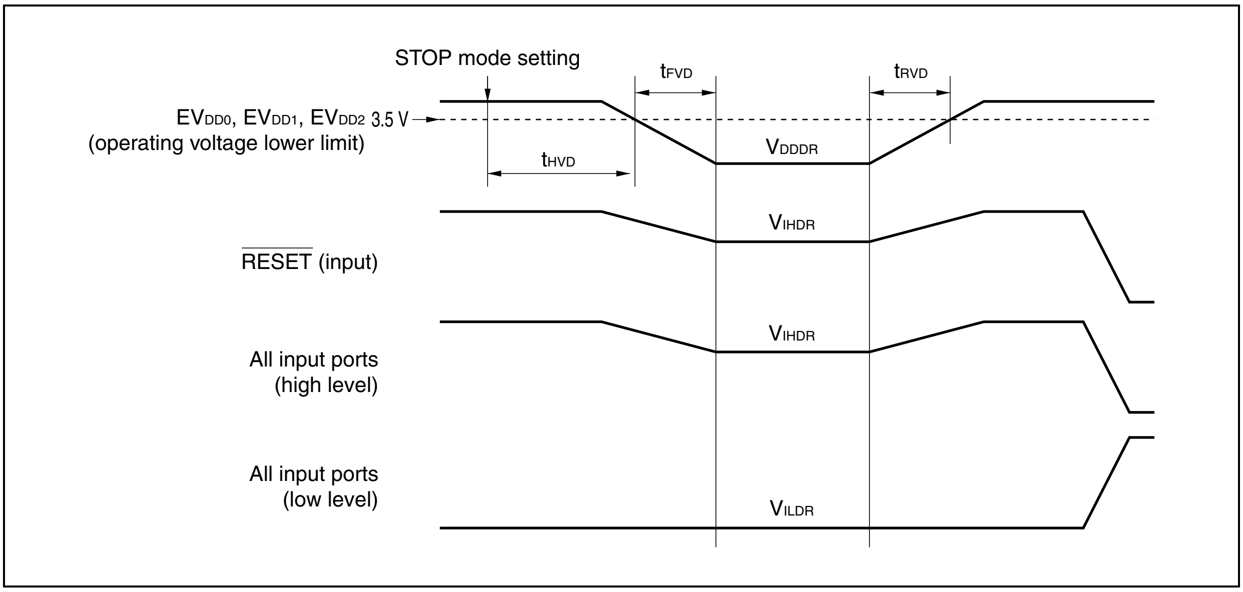
28.1.6 Data retention characteristics

STOP mode (T_A = –40 to +85°C, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode	**Note**		5.5	V
Data retention current	I_{DDDR}	V_{DD0} = V_{DD1} = V_{DD2} = V_{DDDR}		40	800	μA
Supply voltage rise time	t_{RVD}		1			μs
Supply voltage fall time	t_{FVD}		1			μs
Supply voltage hold time (from STOP mode setting)	t_{HVD}		0			ms
Data retention input voltage, high	V_{IHDR}	All input ports	0.9V_{DDDR}		V_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	All input ports	EV_{SS}		0.1V_{DDDR}	V

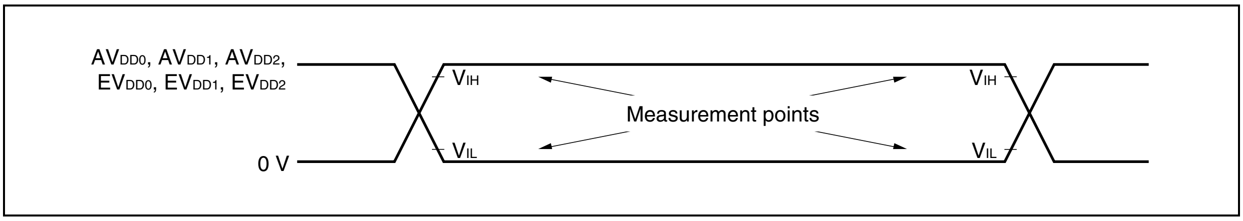
Note When the low-voltage detector (LVI) reset mode is not used (LVIM.LVIMD bit = 0): POC detection voltage (V_{POC0})

When the low-voltage detector (LVI) reset mode is used (LVIM.LVIMD bit = 1): LVI detection voltage (V_{LVI0}/V_{LVI1})

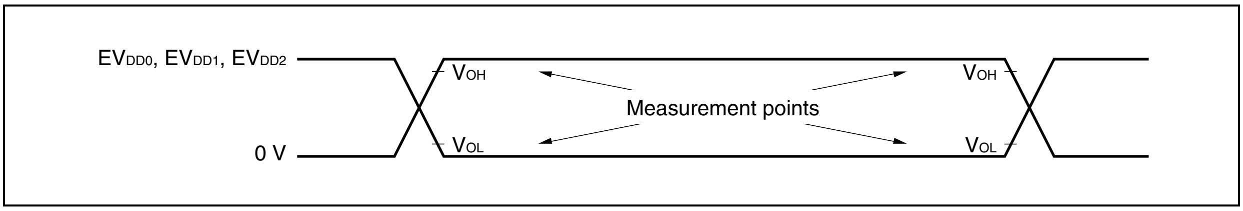

STOP mode setting
t_{FVD}
t_{RVD}
EV_{DD0}, EV_{DD1}, EV_{DD2} 3.5 V
(operating voltage lower limit)
t_{HVD}
V_{DDDR}
V_{IHDR}
RESET (input)
V_{IHDR}
All input ports (high level)
All input ports (low level)
V_{ILDR}

28.1.7 AC characteristics

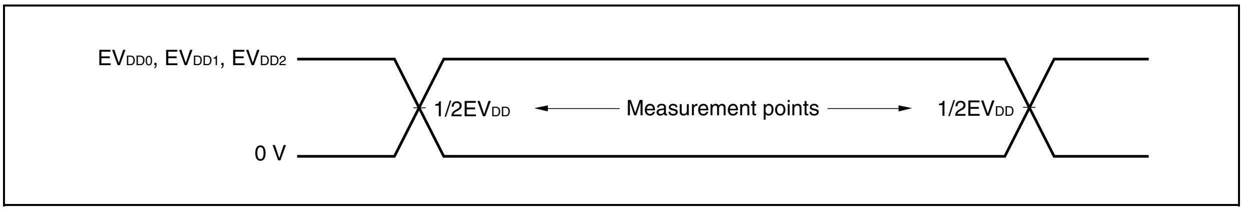
AC test input measurement points (external bus, pins other than CSIF0 to CSIF2)



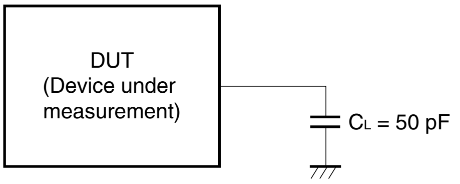
AC test output measurement points (external bus, pins other than CSIF0 to CSIF2)



AC test I/O measurement points (external bus, CSIF0 to CSIF2 pins)



Load conditions

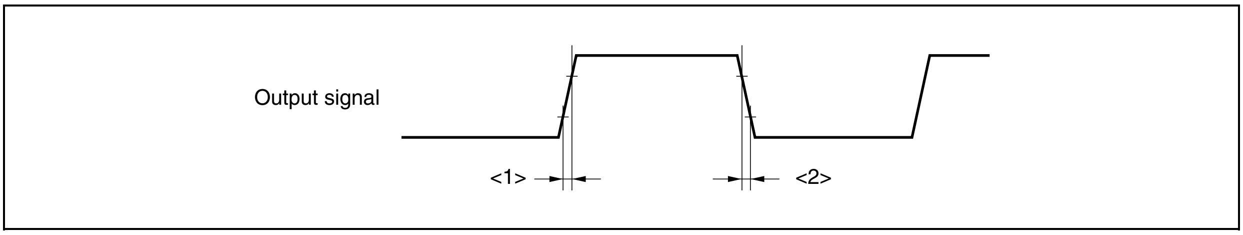


Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

(1) Output signal timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	t_{OR}	<1>	P07		5	ns
			PDL0 to PDL15, DDO		8	ns
			Other than above		15	ns
Output fall time	t_{OF}	<2>	P07		5	ns
			PDL0 to PDL15, DDO		8	ns
			Other than above		15	ns



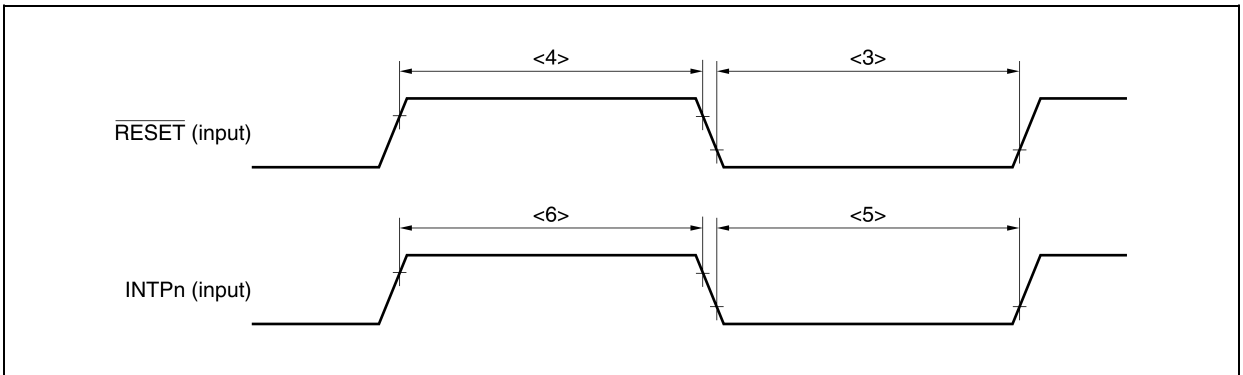
(2) Reset, external interrupt timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<3>	Power is on, STOP mode is released	500 + T_{os}		ns
			Other than above	500		ns
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	<4>		500		ns
INTPn low-level width	t_{WITL}	<5>	n = 00 to 19 (analog noise elimination)	500		ns
			n = 00 to 02, 17 to 19 (digital noise elimination)	$4T_{smp}$		ns
INTPn high-level width	t_{WITH}	<6>	n = 00 to 19 (analog noise elimination)	500		ns
			n = 00 to 02, 17 to 19 (digital noise elimination)	$4T_{smp}$		ns

Remark T_{os}: Oscillation stabilization time
T_{smp}: Noise elimination sampling clock cycle (set by INTNFCn register)

Reset/Interrupt

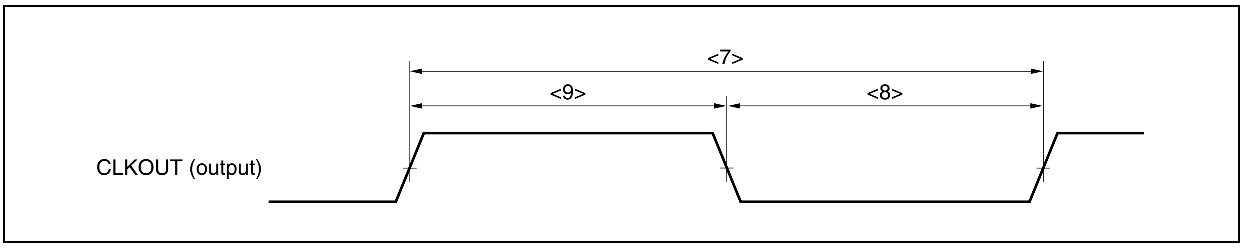


Remark n = 00 to 19

(3) CLKOUT output timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<7>		31.25 ns	3.2 μs	
Low-level width	t_{WKH}	<8>		t_{CYK}/2 – 6.2		ns
High-level width	t_{WKL}	<9>		t_{CYK}/2 – 6.2		ns



(4) Bus timing

(a) Read cycle (CLKOUT asynchronous)

(T_A = −40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = 4.5 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from address to ASTB↓	t_{DAST2}	<10>		$(0.5 + w_{AS})$ T − 5		ns
ASTB high-level width	t_{WSTH}	<11>		$(1 + w_{AS} + i)$ T − 10		ns
Address hold time from ASTB↓	t_{HSTA}	<12>		$(0.5 + w_{AH})$ T − 5		ns
Delay time from address to $\overline{\text{RD}}$↓	t_{DARD2}	<13>		$(1 + w_{AS} + w_{AH})$ T − 10		ns
Delay time from $\overline{\text{RD}}$↓ to address float	t_{FRDA}	<14>			15	ns
Data input setup time from ASTB↓	t_{DSTID}	<15>			$(1.5 + w_D + w + w_{AH})$ T − 10	ns
Data input setup time from $\overline{\text{RD}}$↓	t_{DRDID2}	<16>			$(1 + w_D + w)$ T − 10	ns
Delay time from ASTB↓ to $\overline{\text{RD}}$↓	t_{DSTRD3}	<17>		$(0.5 + w_{AH})$ T − 5		ns
Data input hold time (from $\overline{\text{RD}}$↑)	t_{HRDID2}	<18>		0		ns
Delay time from $\overline{\text{RD}}$↑ to bus output	t_{DRDOD2}	<19>		$(1 + i)$ T − 5		ns
Delay time from $\overline{\text{RD}}$↑ to ASTB↑	t_{DRDST}	<20>		0.5T − 5		ns
$\overline{\text{RD}}$ low-level width	t_{WRDL2}	<21>		$(1 + w_D + w)$ T − 10		ns
$\overline{\text{RD}}$ high-level width	t_{WRDH2}	<22>		$(2 + i + w_{AS} + w_{AH})$ T − 10		ns
High-level hold time from $\overline{\text{RD}}$↑ to $\overline{\text{WRn}}$	t_{HRDWR2}	<23>		$(2 + i + w_{AS} + w_{AH})$ T − 10		ns
$\overline{\text{WAIT}}$ setup time (to address)	t_{DAWT2}	<24>			$(1.5 + w_D + w + w_{AS} + w_{AH})$ T − 10	ns
$\overline{\text{WAIT}}$ hold time (from address)	t_{HAWT2}	<25>		$(1.5 + w_D + w + w_{AS} + w_{AH})$ T		ns
$\overline{\text{WAIT}}$ setup time (to ASTB↓)	t_{DSTWT}	<26>			$(1 + w_D + w + w_{AH})$ T − 10	ns
$\overline{\text{WAIT}}$ hold time (from ASTB↓)	t_{HSTWT}	<27>		$(1 + w_D + w + w_{AH})$ T		ns
$\overline{\text{WAIT}}$ setup time (to $\overline{\text{RD}}$↓)	t_{DRDWT2}	<28>			$(0.5 + w_D + w)$ T − 10	ns
$\overline{\text{WAIT}}$ hold time (from $\overline{\text{RD}}$↓)	t_{HRDWT2}	<29>		$(0.5 + w_D + w)$ T		ns

Cautions 1. Set T in accordance with the following condition.

40 ns ≤ T

2. Be sure to insert the address setup waits and address hold waits.

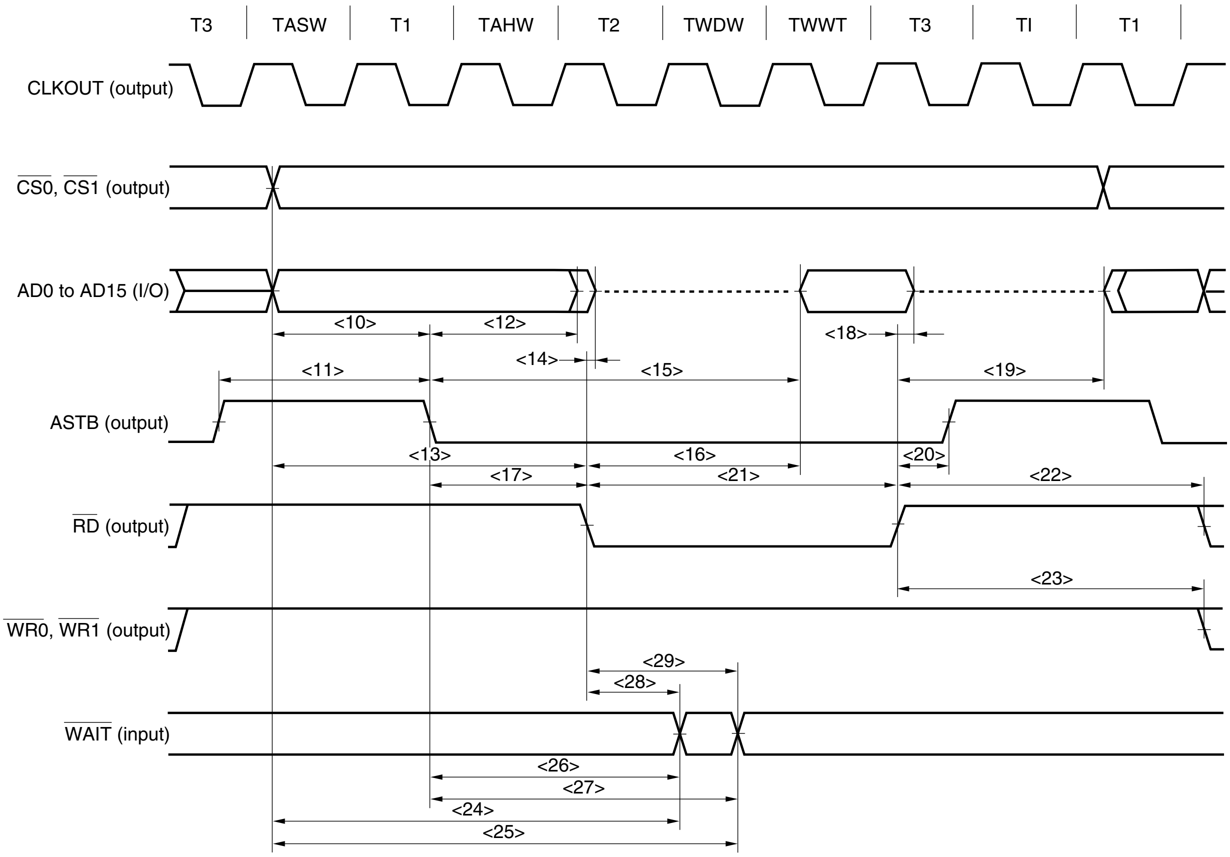
Remarks 1. w_{AS}: Number of address setup waits by the AWC register
w_{AH}: Number of address hold waits by the AWC register
w_D: Number of data waits by the DWC0 register
w: Number of external waits by the $\overline{\text{WAIT}}$ pin

2. T = $1/f_{CPU}$ (f_{CPU}: CPU clock frequency)

3. n = 0, 1

4. i: Number of idle states

Read cycle (CLKOUT asynchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, number of waits by WAIT pin is 1 (when an active level (low level) is input for one cycle during the determined wait period), and number of idle states is 1.

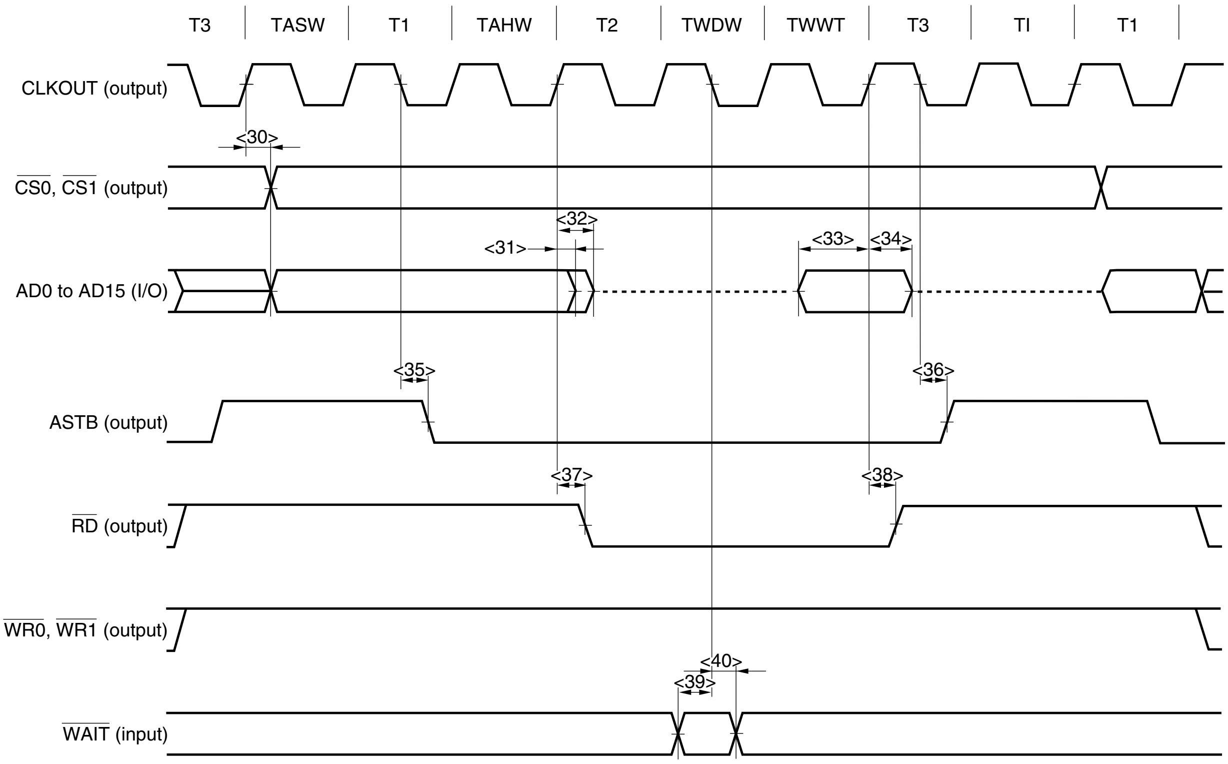
(b) Read cycle (CLKOUT synchronous)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = 4.5 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA2}	<30>			12	ns
Address hold time from CLKOUT↑	t_{HKA3}	<31>		0		ns
Delay time from CLKOUT↑ to address float	t_{FKA}	<32>			12	ns
Data input setup time (to CLKOUT↑)	t_{SIDK2}	<33>		16		ns
Data input hold time (from CLKOUT↑)	t_{HKID2}	<34>		0		ns
Delay time from CLKOUT↓ to ASTB↓	t_{DKST3}	<35>		0	12	ns
Delay time from CLKOUT↓ to ASTB↑	t_{DKST4}	<36>		0	12	ns
Delay time from CLKOUT↑ to $\overline{\text{RD}}$↓	t_{DKRD3}	<37>		0	12	ns
Delay time from CLKOUT↑ to $\overline{\text{RD}}$↑	t_{DKRD4}	<38>		0	12	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↓)	t_{SWTK2}	<39>		16		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↓)	t_{HKWT2}	<40>		0		ns

Caution Be sure to insert the address setup waits and address hold waits.

Read cycle (CLKOUT synchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, number of waits by WAIT pin is 1 (when an active level (low level) is input for one cycle during the determined wait period), and number of idle states is 1.

(c) Write cycle (CLKOUT asynchronous)

(T_A = −40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = 4.5 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from address to ASTB↓	t_{DAST2}	<10>		$(0.5 + w_{AS})$ T − 5		ns
ASTB high-level width	t_{WSTH}	<11>		$(1 + w_{AS} + i)$ T − 10		ns
Address hold time from ASTB↓	t_{HSTA}	<12>		$(0.5 + w_{AH})$ T − 5		ns
Delay time from address to $\overline{\text{WRn}}$↓	t_{DAWR2}	<41>		$(1 + w_{AS} + w_{AH})$ T − 10		ns
Delay time from $\overline{\text{WRn}}$↓ to data output	t_{DWROD3}	<42>			5	ns
Delay time from ASTB↓ to $\overline{\text{WRn}}$↓	t_{DSTWR3}	<43>		$(0.5 + w_{AH})$ T − 5		ns
Delay time from data output to $\overline{\text{WRn}}$↑	t_{DODWR2}	<44>		$(1 + w_D + w)$ T − 10		ns
Data output hold time from $\overline{\text{WRn}}$↑	t_{HWROD2}	<45>		T − 5		ns
Delay time from $\overline{\text{WRn}}$↑ to ASTB↑	t_{DWRST}	<46>		0.5T − 5		ns
$\overline{\text{WRn}}$ low-level width	t_{WWRL2}	<47>		$(1 + w_D + w)$ T − 10		ns
$\overline{\text{WRn}}$ high-level width	t_{WWRH2}	<48>		$(2 + w_{AS} + w_{AH})$ T − 10		ns
High-level hold time from $\overline{\text{WRn}}$↑ to $\overline{\text{RD}}$	t_{HWRRD2}	<49>		$(2 + w_{AS} + w_{AH})$ T − 10		ns
$\overline{\text{WAIT}}$ setup time (to address)	t_{DAWT2}	<24>			$(1.5 + w_D + w + w_{AS} + w_{AH})$ T − 10	ns
$\overline{\text{WAIT}}$ hold time (from address)	t_{HAWT2}	<25>		$(1.5 + w_D + w + w_{AS} + w_{AH})$ T		ns
$\overline{\text{WAIT}}$ setup time (to ASTB↓)	t_{DSTWT}	<26>			$(1 + w_D + w + w_{AH})$ T − 10	ns
$\overline{\text{WAIT}}$ hold time (from ASTB↓)	t_{HSTWT}	<27>		$(1 + w_D + w + w_{AH})$ T		ns
$\overline{\text{WAIT}}$ setup time (to $\overline{\text{WRn}}$↓)	t_{DWRWT2}	<50>			$(0.5 + w_D + w)$ T − 10	ns
$\overline{\text{WAIT}}$ hold time (from $\overline{\text{WRn}}$↓)	t_{HWRWT2}	<51>		$(0.5 + w_D + w)$ T		ns

Cautions 1. Set T in accordance with the following condition.

40 ns ≤ T

2. Be sure to insert the address setup waits and address hold waits.

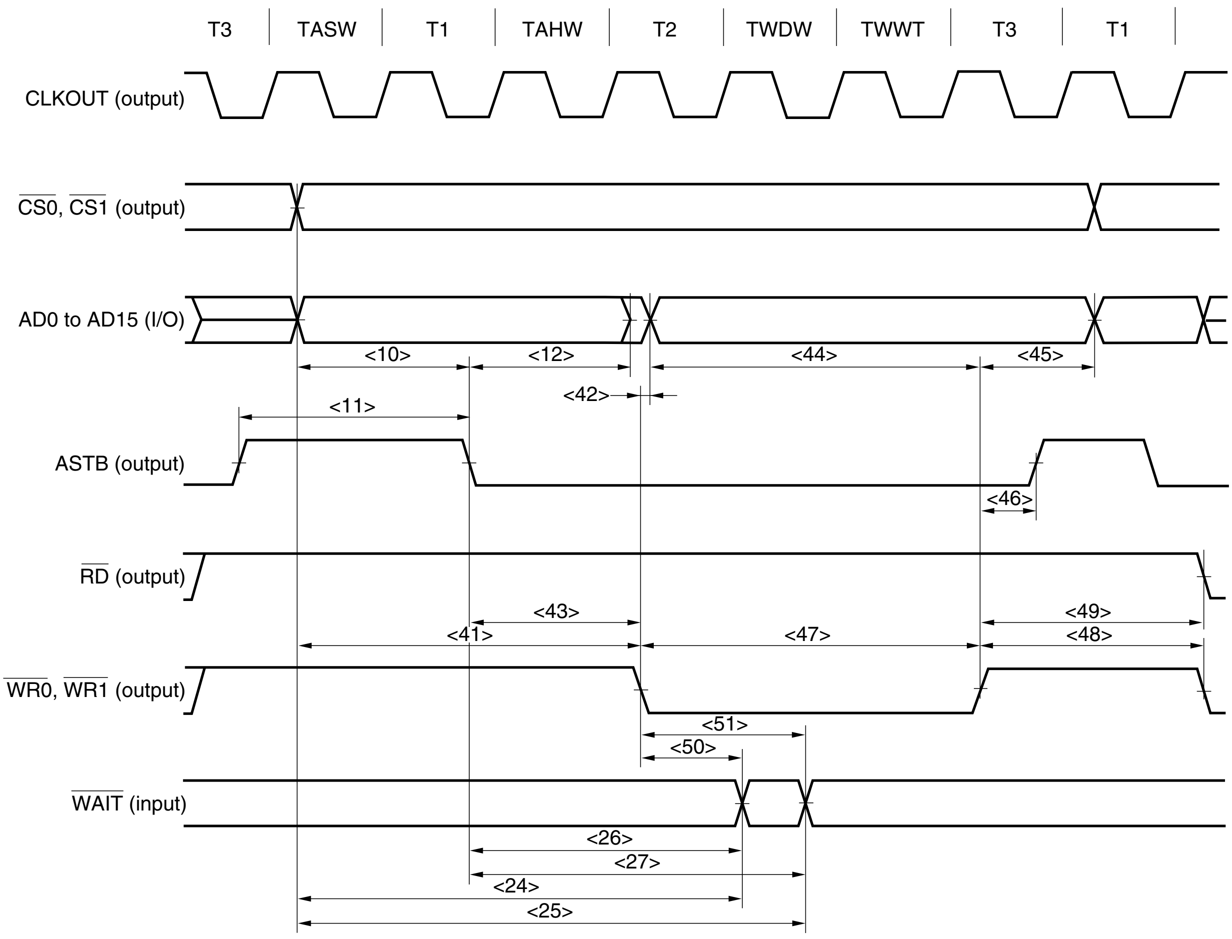
Remarks 1. w_{AS}: Number of address setup waits by the AWC register
w_{AH}: Number of address hold waits by the AWC register
w_D: Number of data waits by the DWC0 register
w: Number of external waits by the $\overline{\text{WAIT}}$ pin

2. T = 1/f_{CPU} (f_{CPU}: CPU operating clock frequency)

3. n = 0, 1

4. i: Number of idle states

Write cycle (CLKOUT asynchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, and number of waits by WAIT pin is 1 (when an active level (low level) is input for one cycle during the determined wait period).

(d) Write cycle (CLKOUT synchronous)

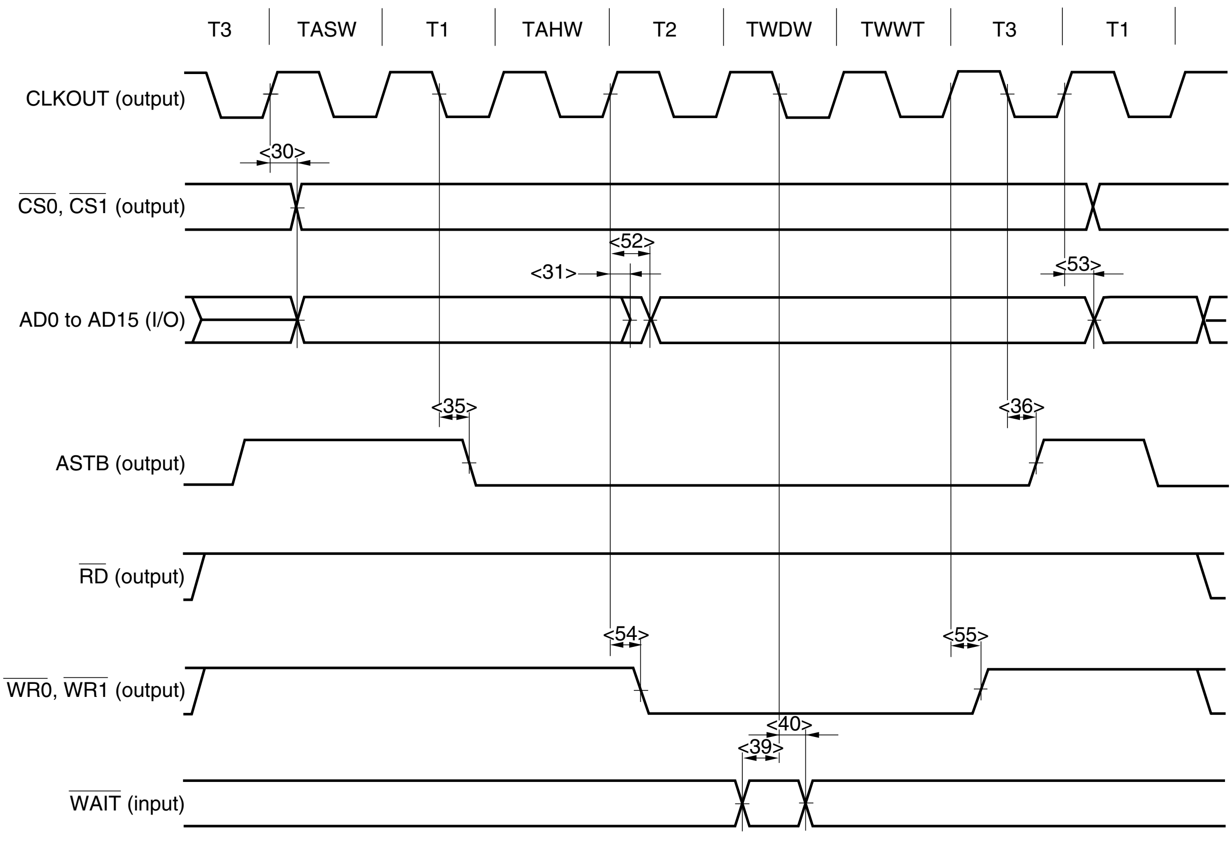
**(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = 4.5 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA2}	<30>			12	ns
Address hold time from CLKOUT↑	t_{HKA3}	<31>		0		ns
Delay time from CLKOUT↓ to ASTB↓	t_{DKST3}	<35>		0	12	ns
Delay time from CLKOUT↓ to ASTB↑	t_{DKST4}	<36>		0	12	ns
Delay time from CLKOUT↑ to data output	t_{DKOD3}	<52>			12	ns
Data output hold time from CLKOUT↑	t_{HKOD2}	<53>		0		ns
Delay time from CLKOUT↑ to $\overline{\text{WRn}}$↓	t_{DKWR3}	<54>		0	12	ns
Delay time from CLKOUT↑ to $\overline{\text{WRn}}$↑	t_{DKWR4}	<55>		0	12	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↓)	t_{SWTK2}	<39>		16		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↓)	t_{HKWT2}	<40>		0		ns

Caution Be sure to insert the address setup waits and address hold waits.

Remark n = 0, 1

Write cycle (CLKOUT synchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, and number of waits by WAIT pin is 1 (when an active level (low level) is input for one cycle during the determined wait period).

(3) Timer timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIn high-level width[Notes 1, 2]	t_{WTIHn}	<56>	n = B00 to B03	12T + 10		ns
			n = A20, A21	$3T_{smp1}$ + 10		ns
TIn low-level width[Notes 1, 2]	t_{WTILn}	<57>	n = B00 to B03	12T + 10		ns
			n = A20, A21	$3T_{smp1}$ + 10		ns
EVTBm high-level width[Note 1]	t_{WEVBHm}	<58>	m = 0, 1	12T + 10		ns
EVTBm low-level width[Note 1]	t_{WEVBLm}	<59>	m = 0, 1	12T + 10		ns
TRGBm high-level width[Note 1]	t_{WTRHm}	<60>	m = 0, 1	12T + 10		ns
TRGBm low-level width[Note 1]	t_{WTRLm}	<61>	m = 0, 1	12T + 10		ns
TENCm0/TENCm1 high-level width[Note 3]	t_{WENCHm}	<62>	m = 0, 1	$3T_{smp2}$ + 10		ns
TENCm0/TENCm1 low-level width[Note 3]	t_{WENCLm}	<63>	m = 0, 1	$3T_{smp2}$ + 10		ns
TECRm high-level width[Note 3]	t_{WCRHm}	<64>	m = 0, 1	$3T_{smp2}$ + 10		ns
TECRm low-level width[Note 3]	t_{WCRLm}	<65>	m = 0, 1	$3T_{smp2}$ + 10		ns
TITk0/TITk1 high-level width[Note 3]	t_{WTITHk}	<66>	k = 0 to 3	$3T_{smp2}$ + 10		ns
TITk0/TITk1 low-level width[Note 3]	t_{WTITLk}	<67>	k = 0 to 3	$3T_{smp2}$ + 10		ns
EVTTm high-level width[Note 3]	t_{WEVTHm}	<68>	m = 0, 1	$3T_{smp2}$ + 10		ns
EVTTm low-level width[Note 3]	t_{WEVTLm}	<69>	m = 0, 1	$3T_{smp2}$ + 10		ns
TENCm0/TENCm1 input time differential[Note 3]	t_{PHUDm}	<70>	m = 0, 1	$3T_{smp2}$ + 10		ns

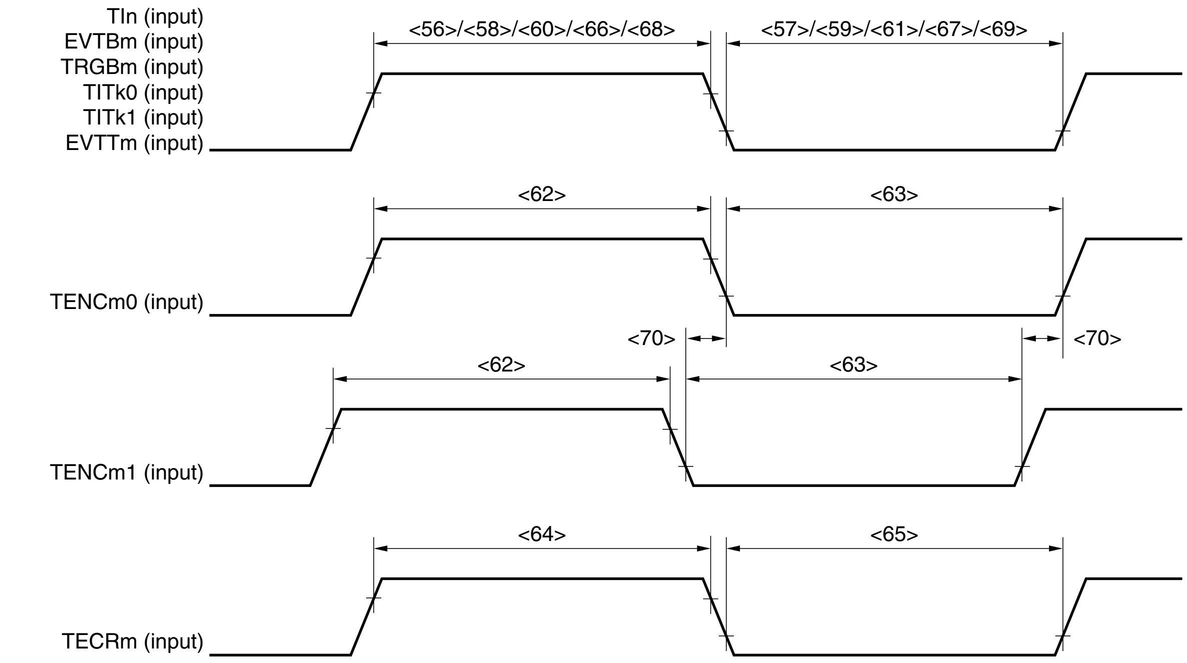
Notes 1. T = 1/fxx

2. T_{smp1}: Noise elimination sampling clock cycle (set by the TANFC2 register)

3. T_{smp2}: Noise elimination sampling clock cycle (set by the TTNFC0 to TTNFC3 registers)

Remark The above specification shows a pulse width that is accurately detected as a valid edge. Even if a pulse narrower than the above specification is input, therefore, it may be detected as a valid edge.

Timer Input Timing



Remark n = A20, A21, B00 to B03
k = 0 to 3
m = 0, 1

(4) UARTA timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transmission rate				1.25	Mbps

(5) UARTB timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transmission rate				5.00	Mbps

(6) CSIF timing

(a) Master mode

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKFn}}$ cycle	t_{KCYM}	<71>		125		ns
$\overline{\text{SCKFn}}$ high-/low-level width	t_{KWHM}, t_{KWLM}	<72>		t_{KCYM}/2 – 10		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↑)	t_{SSIM}	<73>		30		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↓)				30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSIM}	<74>		30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↓)				30		ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↓)	t_{DSOM}	<75>			30	ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↑)					30	ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSOM}	<76>		t_{KCYM}/2 – 10		ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↓)				t_{KCYM}/2 – 10		ns

Remark n = 0 to 2

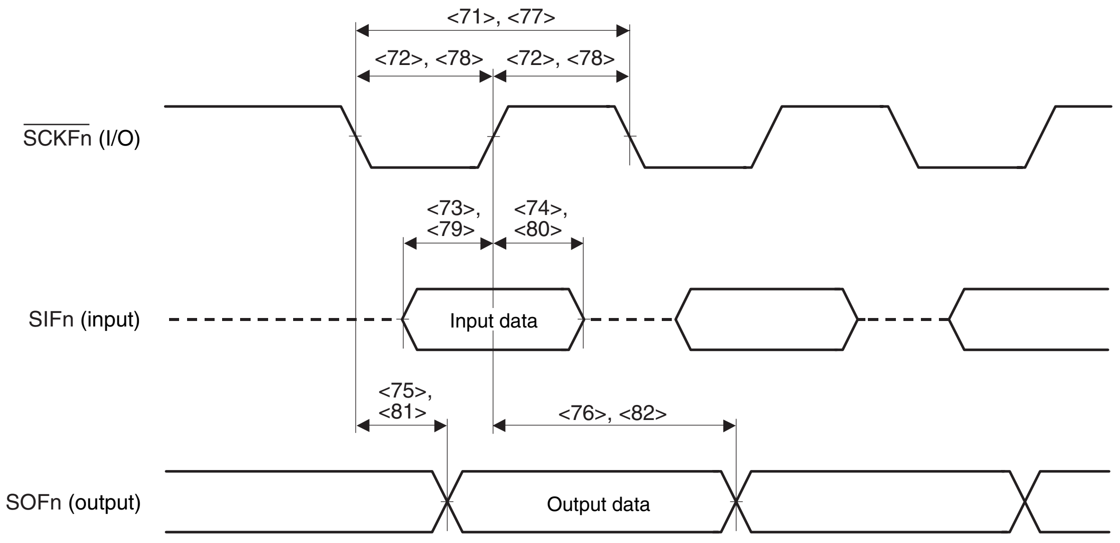
(b) Slave mode

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} =1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKFn}}$ cycle	t_{KCYS}	<77>		125		ns
$\overline{\text{SCKFn}}$ high-/low-level width	t_{KWHS}, t_{KWLS}	<78>		t_{KCYS}/2 – 10		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↑)	t_{SSIS}	<79>		30		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↓)				30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSIS}	<80>		30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↓)				30		ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↓)	t_{DSOS}	<81>			30	ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↑)					30	ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSOS}	<82>		t_{KCYS}/2 – 10		ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↓)				t_{KCYS}/2 – 10		ns

Remark n = 0 to 2

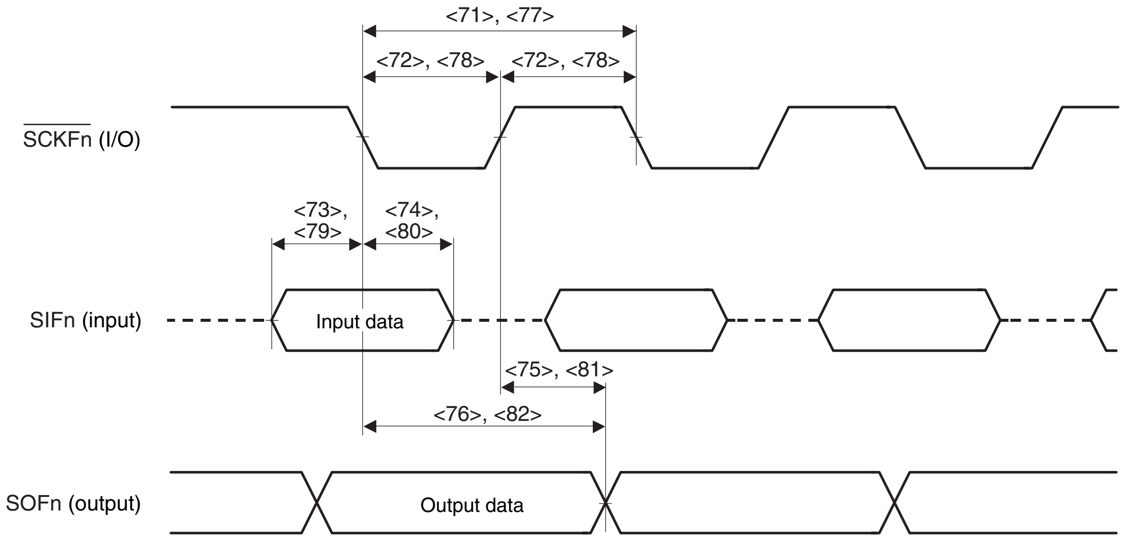
CSIF timing when CFnCKP and CFnDAP bits of CFnCTL1 register = 00



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

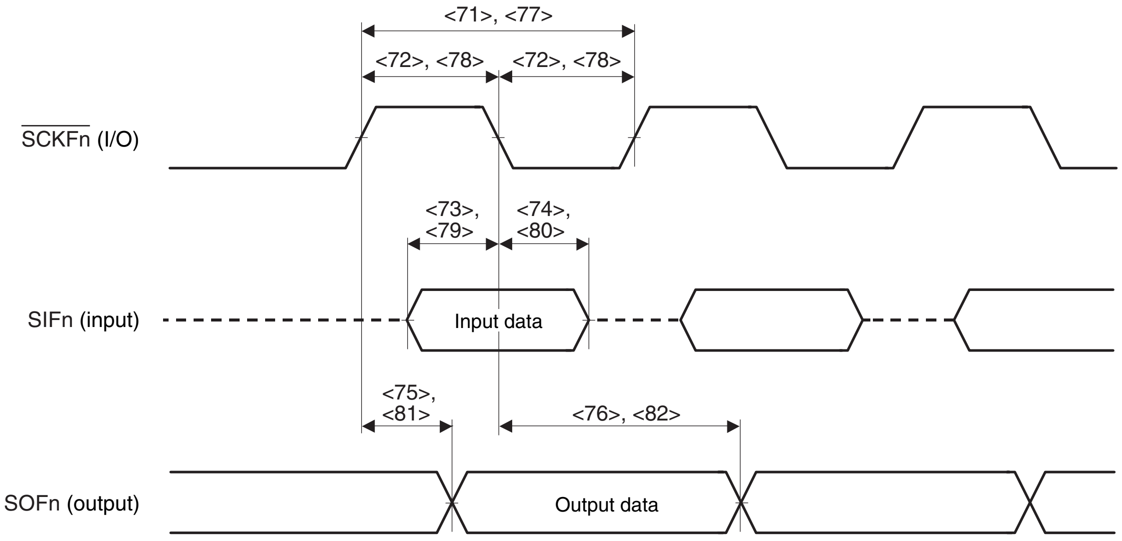
CSIF timing when CFnCKP and CFnDAP bits of CFnCTL1 register = 01



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

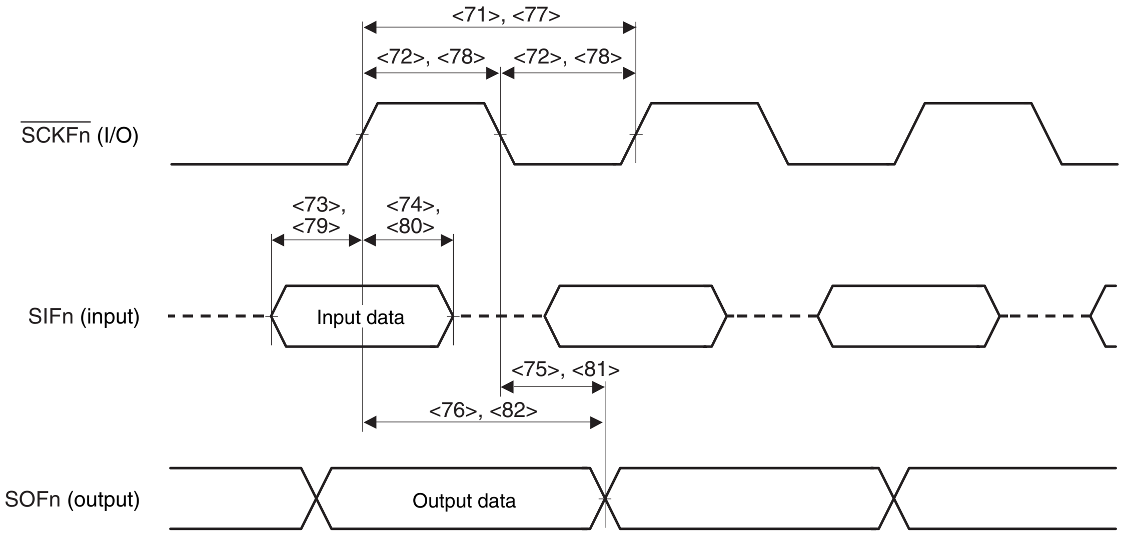
CSIF timing when CFnCKP and CFnDAP bits of CFnCTL1 register = 10



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

CSIF timing when CFnCKP and CFnDAP bits of CFnCTL1 register = 11



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

(7) I^{2}C bus timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter		Symbol		Standard Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL clock frequency		f_{CLK}	–	0	100	0	400	kHz
Bus free time (between stop condition and start condition)		t_{BUF}	<83>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<84>	4.0	–	0.6	–	μs
SCL clock low-level width		t_{LOW}	<85>	4.7	–	1.3	–	μs
SCL clock high-level width		t_{HIGH}	<86>	4.0	–	0.6	–	μs
Start/restart condition setup time		$t_{SU:STA}$	<87>	4.7	–	0.6	–	μs
Data hold time	CBUS-compatible master	$t_{HD:DAT}$	<88>	5.0	–	–	–	μs
	I^{2}C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<89>	250	–	100[Note 4]	–	ns
SDA, SCL signal rise time		t_R	<90>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA, SCL signal fall time		t_F	<91>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<92>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<93>	–	–	0	50	ns
Each bus line capacitive load		Cb	–	–	400	–	400	pF

Notes 1. The first clock pulse is generated after a hold time during the start condition.

2. The system must internally supply a hold time of at least 300 ns for the SDA signal (at $V_{IHmin.}$ of SCL signal) to fill the undefined area at the falling edge of SCL.

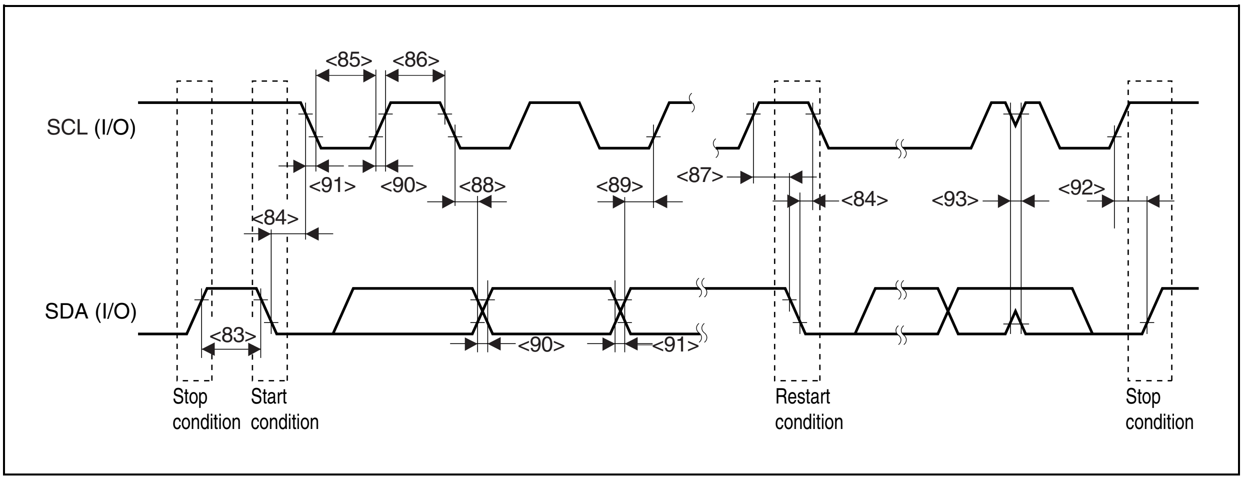
3. If the system does not extend the low hold time (t_{LOW}) of the SCL signal, the maximum data hold time ($t_{HD:DAT}$) must be satisfied.

4. The high-speed mode I^{2}C bus can be used in the standard mode I^{2}C bus system. In this case, make sure that the following conditions are satisfied.

- If system does not extend the low hold time of the SCL signal
  $t_{SU:\,DAT} \geq 250$ ns
- If system extends the low hold time of SCL signal
  Sends the next data bit to the SDA line before the SCL line is released ($t_{Rmax.}$ + $t_{SU:DAT}$ = 1000 + 250 = 1250 ns: standard mode I^{2}C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C bus timing



(8) High-impedance control timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V, EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Oscillation stop → timer output high impedance	t_{CLM}	When clock monitor is operating		65	μs
Input to TOB0OFF, TOB01OFF → timer output high impedance	t_{HTQn}			300	ns
Input to TOT2OFF → timer output high impedance	t_{HTPm}			300	ns
Input to ANI00/ANI05 to ANI02/ANI07 → timer output high impedance	t_{ANI0}			10	μs

28.1.8 Characteristics of A/D converters 0 and 1

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V, EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = AV_{REFP0} = AV_{REFP1} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			12	12	12	bit
Overall error[Note 1]					±10	LSB
Conversion time	t_{CONV}		2.00			μs
					8.00	μs
Zero scale error[Note 1]					±10	LSB
Full-scale error[Note 1]					±10	LSB
Integral linearity error[Note 1]					±4	LSB
Differential linearity error[Note 1]					±2.5	LSB
Analog reference voltage	AV_{REF}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current[Note 2]	AI_{DD}	Operating		4.5	7.5	mA
	AI_{DDS}	In STOP mode[Note 3]		3.5	17.5	μs

Notes 1. Excludes quantization error (±0.5 LSB).

2. This value is for only one A/D converter (A/D converter 0 or 1).

3. Stop A/D converters 0 and 1 (ADnSCM.ADnCE bit = 0) before setting STOP mode.

Remarks 1. LSB: Least Significant Bit

2. f_{AD01}: Base clock for A/D converters 0 and 1

3. n = 0, 1

28.1.9 Characteristics of A/D converter 2

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note 1]					±4.0	LSB
Conversion time	t_{CONV}		3.00		10.00	μs
Zero scale error[Note 1]					±4.0	LSB
Full-scale error[Note 1]					±4.0	LSB
Integral linearity error[Note 1]					±4.0	LSB
Differential linearity error[Note 1]					±2.0	LSB
Analog reference voltage	AV_{REF}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current	AI_{DD}	During operation		3.5	7	mA
	AI_{DDS}	In STOP mode[Note 2]		1	10	μA

Notes 1. Excludes quantization error (±0.5 LSB).

2. Stop A/D converter 2 (AD2M0.AD2CE bit = 0) before setting STOP mode.

Remark LSB: Least Significant Bit

28.1.10 Operational amplifier characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IO}				±9.0		mV
Input voltage range	V_I	Gain = 2.500		0.04AV_{DD}		0.36AV_{DD}	V
		Gain = 5.000		0.02AV_{DD}		0.18AV_{DD}	V
		Gain = 10.00		0.01AV_{DD}		0.085AV_{DD}	V
Slew rate[Note 1]	S_R			10	15		V/μs
Gain error		**Note 2**	Gain = 2.500 to 4.444		±1.0	±1.3	%
			Gain = 5.000 to 6.667		±1.0	±1.5	%
			Gain = 8.000, 10.00		±1.0	±1.7	%
		Note 3	Gain = 2.500 to 4.444		±1.0	±2.0	%
			Gain = 5.000 to 6.667		±1.0	±2.1	%
			Gain = 8.000, 10.00		±1.0	±2.2	%
Operating current[Note 4]	I_{OPDD}	During operation			1.8	2.6	mA
	A_{IDDS}	In STOP mode[Note 5]			1.0	10	μA

Notes 1. Inclination characteristic of output voltage from 10% to 90%

2. 4.5 V ≤ AV_{DD0} = AV_{DD1} ≤ 5.5 V

3. 4.0 V ≤ AV_{DD0} = AV_{DD1} < 4.5 V

4. Six operational amplifiers are provided in total. The value shows the operating current per operational amplifier.

5. Stop the operational amplifier (OPnCTL0.OPn2EN to OPn0EN bits = 000) before setting the STOP mode.

Remark Power supplies AV_{DD0} and AV_{DD1} are used for the operational amplifier.

28.1.11 Comparator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V, EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IO}			±3.0		mV
Input voltage range	V_I		AV_{SS}		AV_{DD}	V
Response time	t_{CR}	Input amplitude = 100 mV, at rising edge[Note 1]		1.0		μs
	t_{CF}	Input amplitude = 100 mV, at falling edge[Note 2]		1.0		μs
Operating current[Note 3]	I_{CPDD}	During operation			250	μA
	A_{IDDS}	In STOP mode[Note 4]		2.0	20	nA
Resolution of D/A converter for reference voltage generator	R_{ES}			8		bit
Overall error of D/A converter for reference voltage generator	A_{INL}	R_{LOAD} ≥ 4 MΩ			±1.2	%FSR
Operation current of D/A converter for reference voltage generator[Note 3]	I_{DADD}	During operation			5	mA
	A_{IDDS2}	In STOP mode[Note 4]			10	μA

Notes 1. Characteristics of pulse response when ANIm input changes from the comparator reference voltage – 100 mV to the comparator reference voltage + 100 mV

2. Characteristics of pulse response when ANIm input changes from the comparator reference voltage + 100 mV to the comparator reference voltage – 100 mV

3. Six comparators are provided in total. The value shows the operating current per comparator.

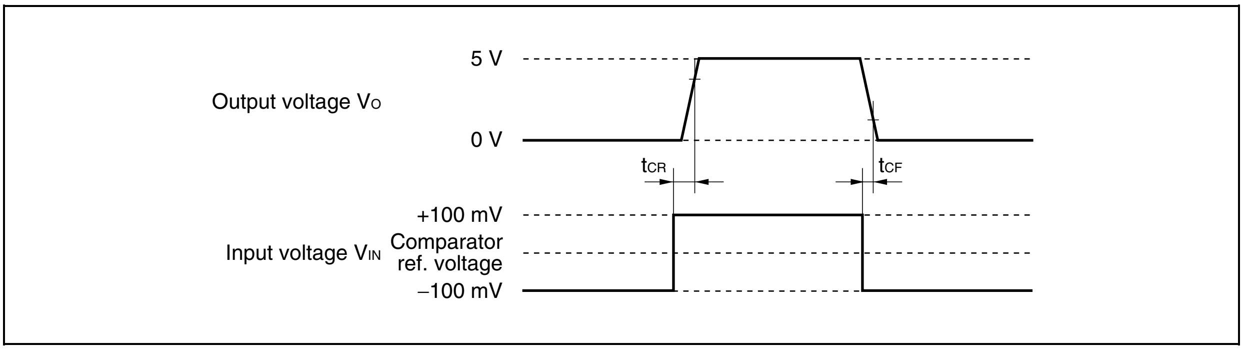
4. Stop the comparator (CMPnCTL0 register = 00H) before setting STOP mode.

Remarks 1. Power supplies for the comparators are AV_{DD0} and AV_{DD1}.

2. m = 05 to 07, 15 to 17
n = 0, 1

3. R_{LOAD}: Total value of ladder resistor (see **Figures 12-3** and **12-4**.)

Comparator Characteristics



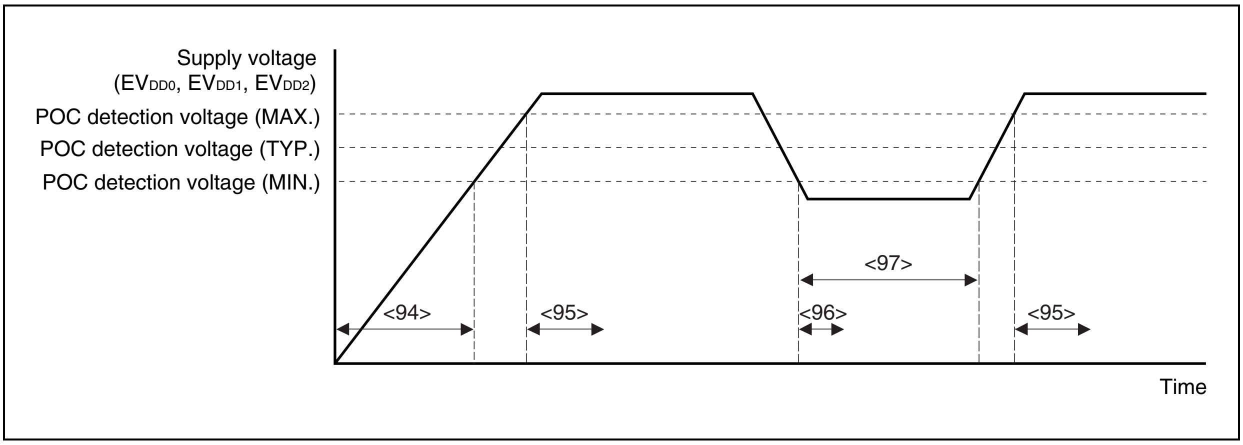
28.1.12 Power-on-clear circuit (POC)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
POC detection voltage	V_{POC0}			3.5	3.7	3.9	V
Supply voltage rise time	t_{PTH}	<94>	EV_{DD0}, EV_{DD1}, and EV_{DD2} = 0 to 3.5 V	2.5 μs		1.8 s	
Response time 1[Note 1]	t_{PTHD}	<95>	After EV_{DD0}, EV_{DD1}, and EV_{DD2} reach 3.9 V on power application			3.0	ms
Response time 2[Note 2]	t_{PD}	<96>	After EV_{DD0}, EV_{DD1}, and EV_{DD2} drop to 3.5 V on power off			1.0	ms
Minimum width of EV_{DD0}, EV_{DD1}, and EV_{DD2}	t_{PW}	<97>		0.2			ms

Notes 1. The time required to release a reset signal (POCRES) after the POC detection voltage is detected.

2. The time required to output a reset signal (POCRES) after the POC detection voltage is detected.

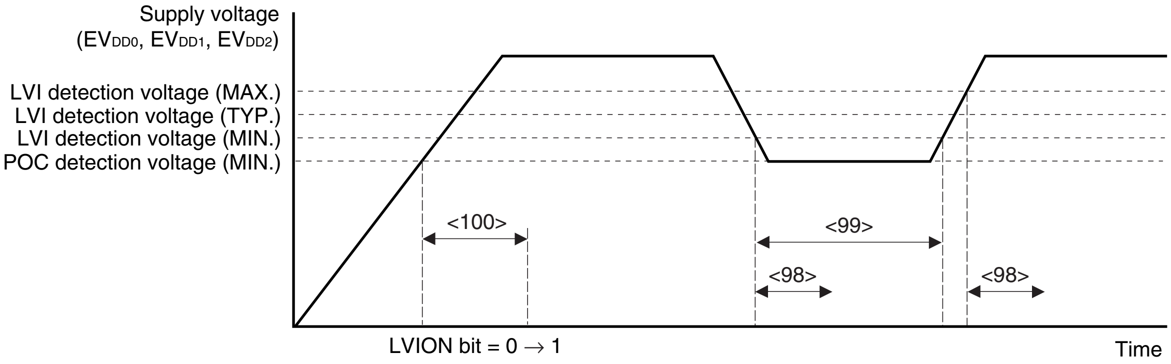

Supply voltage
(EV_{DD0}, EV_{DD1}, EV_{DD2})
POC detection voltage (MAX.)
POC detection voltage (TYP.)
POC detection voltage (MIN.)
<97>
<94>
<95>
<96>
<95>
Time

28.1.13 Low-voltage detector (LVI)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
LVI detection voltage	V_{LVI0}		LVIS.LVIS0 bit = 0	4.2	4.4	4.6	V
	V_{LVI1}		LVIS.LVIS0 bit = 1	4.0	4.2	4.4	V
Response time 1[Note]	t_{LD}	<98>	After EV_{DD0}, EV_{DD1}, and EV_{DD2} reach V_{LVI0}/V_{LVI1} (MAX.) or drop to V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum width of EV_{DD0}, EV_{DD1}, and EV_{DD2}	t_{LW}	<99>		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	<100>	After EV_{DD0}, EV_{DD1}, and EV_{DD2} reach POC detection voltage (MIN.) and the LVIM.LVION bit is changed from 0 to 1		0.1		ms

Note The time required to output an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) after the LVI detection voltage is detected.


Supply voltage
(EV_{DD0}, EV_{DD1}, EV_{DD2})
LVI detection voltage (MAX.)
LVI detection voltage (TYP.)
LVI detection voltage (MIN.)
POC detection voltage (MIN.)
<100>
<99>
<98>
<98>
LVION bit = 0 → 1
Time

28.1.14 Supply voltage application/cutoff timing

(T$_{A}$ = –40 to +85°C, V$_{DD0}$ = V$_{DD1}$ = V$_{DD2}$ = 1.35 to 1.65 V,
EV$_{DD0}$ = EV$_{DD1}$ = EV$_{DD2}$ = AV$_{DD0}$ = AV$_{DD1}$ = AV$_{DD2}$ = 3.5 to 5.5 V, UV$_{DD}$ = 3.0 to 3.6 V,
V$_{SS0}$ = V$_{SS1}$ = V$_{SS2}$ = EV$_{SS0}$ = EV$_{SS1}$ = EV$_{SS2}$ = EV$_{SS4}$ = AV$_{SS0}$ = AV$_{SS1}$ = AV$_{SS2}$ = 0 V, C$_{L}$ = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Delay time from EV$_{DD}$ rise to V$_{DD}$ rise	t$_{RVR}$	When using an external reset	–50	50	ms
		When using an internal reset	–50	0	ms
Delay time from EV$_{DD}$ rise to AV$_{DD}$ rise	t$_{RAR}$	When using an external reset	–50	50	ms
		When using an internal reset	–50	0	ms
Delay time from EV$_{DD}$ rise to UV$_{DD}$ rise	t$_{RUR}$	When using an external reset	–50	50	ms
		When using an internal reset	–50	0	ms
Delay time from EV$_{DD}$ rise to $\overline{\text{RESET}}$ rise	t$_{RRR}$	When using an external reset	T$_{osc}$ + 0.5		ms
Delay time from EV$_{DD}$ fall to V$_{DD}$ fall	t$_{FVF}$		0	50	ms
Delay time from EV$_{DD}$ fall to AV$_{DD}$ fall	t$_{FAF}$		0	50	ms
Delay time from EV$_{DD}$ fall to UV$_{DD}$ fall	t$_{FUF}$		0	50	ms

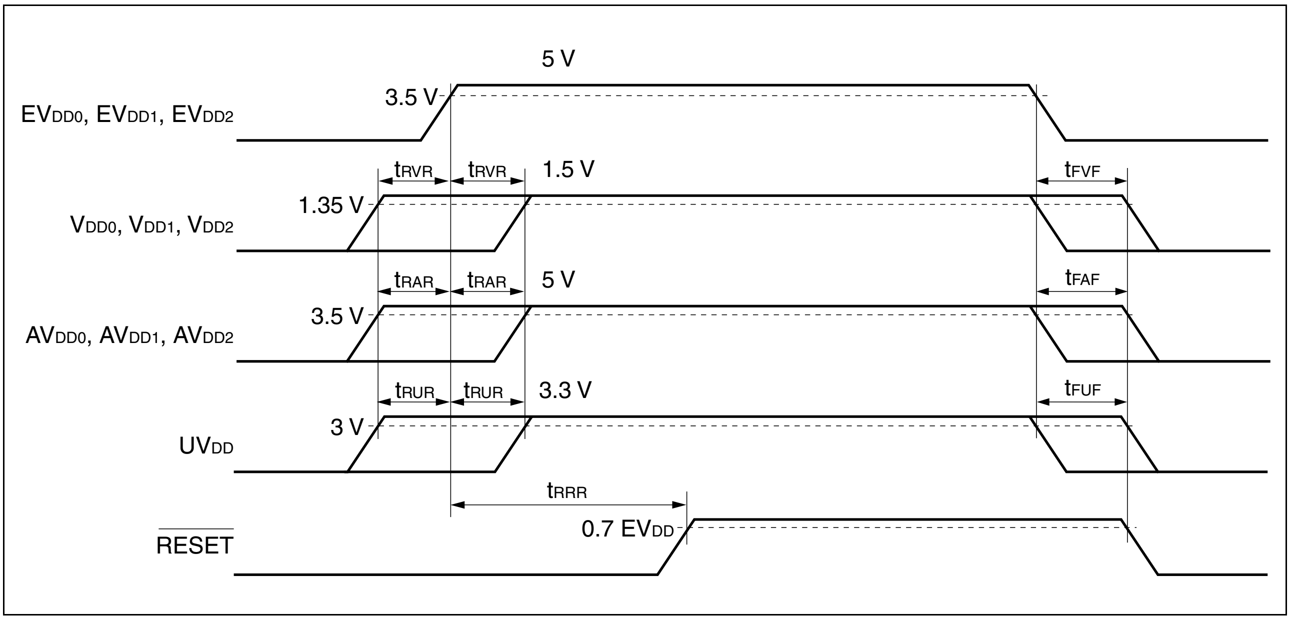
Remark Tosc: Oscillation stabilization time

Supply voltage application/cutoff timing

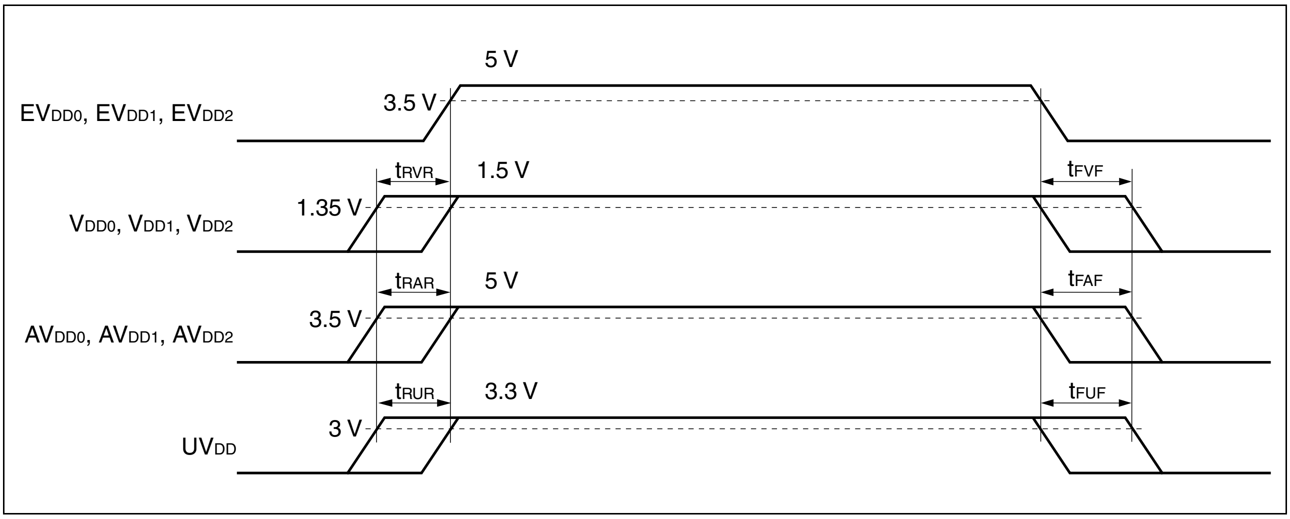
<R> **Cautions 1. There are no regulations for the voltage level and time of EV$_{DD0}$, EV$_{DD1}$, EV$_{DD2}$, V$_{DD0}$, V$_{DD1}$, V$_{DD2}$, AV$_{DD0}$, AV$_{DD1}$, AV$_{DD2}$, and UV$_{DD}$ in the process of natural discharge after power supply cutoff.**

2. Apply all of the EV$_{DD0}$, EV$_{DD1}$, EV$_{DD2}$, V$_{DD0}$, V$_{DD1}$, V$_{DD2}$, AV$_{DD0}$, AV$_{DD1}$, AV$_{DD2}$, and UV$_{DD}$ power supplies.
It is prohibited to apply one of these power supplies without supplying them all.

(a) External $\overline{\text{RESET}}$ (recommended conditions)



(b) Internal $\overline{\text{RESET}}$ (recommended conditions)



28.1.15 Flash memory programming characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Rewrite count	C_{ERWR}	**Note**	100			Times

Note Rewrite as follows.

Example when three rewrites: Shipped product →E→P→E→P→E→P (P: Write, E: Erase)

28.2 V850E/IH4-H

28.2.1 Absolute maximum ratings

(T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.5 to +2.0	V
	V_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		–0.5 to +0.5	V
	EV_{DD}			–0.5 to +6.5	V
	EV_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		–0.5 to +0.5	V
	FV_{DD}			–0.5 to +6.5	V
	AV_{DD}			–0.5 to +6.5	V
	AV_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		–0.5 to +0.5	V
	UV_{DD}			–0.5 to +4.6	V
Input voltage	V_{I1}	**Note 1**		–0.5 to EV_{DD} + 0.5 Note 2	V
	V_{I2}	X1, X2		–0.5 to V_{DD} + 0.35	V
Output current, low	I_{OL}	All pins	Per pin	4	mA
			Total of all pins	63	mA
Output current, high	I_{OH}	All pins	Per pin	–4	mA
			Total of all pins	–63	mA
Analog input voltage	V_{IAN}	**Note 3**		–0.5 to AV_{DD} + 0.5 Note 2	V
Analog reference input voltage	V_{IREF}	AV_{REFP0}, AV_{REFP1}		–0.5 to AV_{DD} + 0.5 Note 2	V
Operating ambient temperature	T_A	In normal operating mode		–40 to +85	°C
		In flash memory programming mode		–40 to +85	°C
Storage temperature	T_{stg}			–40 to +125	°C

Notes 1. P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P44, P50 to P56, P70 to P711, P90 to P97, PDL0 to PDL15, $\overline{\text{RESET}}$, FLMD0, $\overline{\text{DRST}}$

2. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

3. P70/ANI20 to P711/ANI211, ANI00/ANI05 to ANI02/ANI07, ANI03, ANI10/ANI15 to ANI12/ANI17, ANI13

Cautions 1. Do not directly connect the output pins (or I/O pins in the output state) of IC products to other output pins (including I/O pins in the output state), power supply pins such as V_{DD} and EV_{DD}, or GND pin. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Remark a = 0 to 2
b = 0 to 4
k = 0 to 2

28.2.2 Capacitance

(T_A = 25°C, V_{DD0} = V_{SS0} = V_{DD1} = V_{SS1} = V_{DD2} = V_{SS2} = EV_{DD0} = EV_{SS0} = EV_{DD1} = EV_{SS1} = EV_{DD2} = EV_{SS2} = EV_{DD3} = EV_{SS3} = EV_{SS4} = FV_{DD} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = AV_{DD2} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	fc = 1 MHz Unmeasured pins returned to 0 V	**Note 1**			15	pF
I/O capacitance	C_{IO}		**Note 2**			15	pF
Output capacitance	C_O		**Note 3**			15	pF

Notes 1. ANI00/ANI05 to ANI02/ANI07, ANI03, ANI10/ANI15 to ANI12/ANI17, ANI13, $\overline{\text{RESET}}$

2. P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P44, P50 to P56, P70 to P711, P90 to P97, PDL0 to PDL15

3. DDO, TRCCLK, TRCDATA0 to TRCDATA3, TRCEND

Cautions 1. Excludes the FLMD0, $\overline{\text{DRST}}$, X1, and X2 pins.

2. In addition to input capacitance, sampling capacitance is added to the ANI00/ANI05 to ANI02/ANI07, ANI03, ANI10/ANI15 to ANI12/ANI17, ANI13, and ANI20 to ANI211 pins when sampling.

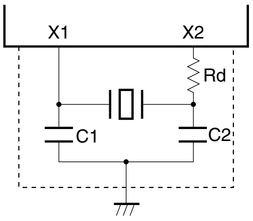
28.2.3 Operating conditions

(T_A = –40 to +85°C, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{XX}	PLL mode	80		100	MHz
		Clock through mode	10		12.5	MHz
CPU clock frequency	f_{CPU}	PLL mode	10		100	MHz
		Clock through mode	1.25		12.5	MHz
V_{DD} voltage	V_{DD}		1.35		1.65	V
EV_{DD} voltage	EV_{DD}		3.0		5.5	V
FV_{DD} voltage	FV_{DD}		4.0		5.5	V
AV_{DD} voltage	AV_{DD}	When A/D converters 0 to 2 are operating	4.0		5.5	V
		When A/D converters 0 to 2 are not operating	3.5		5.5	V
UV_{DD} voltage	UV_{DD}		3.0		3.6	V

28.2.4 Clock oscillator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = 4.0 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic /crystal resonator		Oscillation frequency (f_X)		10		12.5	MHz
		Oscillation stabilization time	After reset release		$2^{15}/f_X$		ms
			After STOP mode release		**Note**		ms

Note The value varies depending on the setting of the oscillation stabilization time select register (OSTS).

Cautions 1. Connect the oscillator as close to the X1 and X2 pins as possible.

2. Do not cross the wiring with the other signal lines in the area enclosed by the broken lines in the above figure.

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

4. Inputting an external clock to the V850E/IH4-H is prohibited.

28.2.5 DC characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = 4.0 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V) (1/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**			0.7EV_{DD}		EV_{DD}	V
	V_{IH2}	**Note 2**			0.8EV_{DD}		EV_{DD}	V
	V_{IH3}	**Note 3**			2.2		EV_{DD}	V
	V_{IH4}	**Note 4**			0.7AV_{DD}		AV_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**			EV_{SS}		0.3EV_{DD}	V
	V_{IL2}	**Note 2**			EV_{SS}		0.2EV_{DD}	V
	V_{IL3}	**Note 3**			EV_{SS}		0.8	V
	V_{IL4}	**Note 4**			AV_{SS}		0.3AV_{DD}	V
Input leakage current, high	I_{LIH1}	V_I = **Note 5**, **Note 6**	Other than X1				5	μA
	I_{LIH2}		X1				20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	Other than X1				–5	μA
	I_{LIL2}		X1				–20	μA
Output leakage current, high	I_{LOH}	V_O = **Note 5**					5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V					–5	μA
Output voltage, high	V_{OH1}	**Note 7**	I_{OH} = –1.0 mA	Total of pins = –57 mA	EV_{DD} – 1.0			V
Output voltage, low	V_{OL1}	**Note 7**	I_{OL} = 1.0 mA	Total of pins = 57 mA			0.4	V
Pull-up resistor	R_{L1}				10	30	120	kΩ
Pull-down resistor[Note 8]	R_{L2}				10	30	120	kΩ

Notes 1. P17, P33, P36, P41, P54 to P56, P90 to P97, PDL0 to PDL15 pins

2. P00 to P07, P10 to P16, P20 to P27, P30 to P32, P34, P35, P37, P40, P42 to P44, P50 to P53, $\overline{\text{RESET}}$, FLMD0 pins

3. $\overline{\text{DRST}}$, DDI, DCK, DMS pins

4. P70 to P711 pins

5. AV_{DD0} = AV_{DD1} = AV_{DD2} = EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3}

6. Except for $\overline{\text{DRST}}$ pin

7. P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P44, P50 to P56, P90 to P97, PDL0 to PDL15, DDO, TRCCLK, TRCDATA0 to TRCDATA3, TRCEND pins

8. $\overline{\text{DRST}}$ pin only

Remark The characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V, EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = 4.0 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.[Note 1]	MAX.	Unit
V_{DD} supply current[Note 2]	I_{DD1}	f_{XX} = 100 MHz	Normal operation		125	205	mA
	I_{DD2}		HALT mode		66	143	mA
	I_{DD3}		IDLE mode		6	50	mA
	I_{DD4}	STOP mode			0.1	16	mA

Notes 1. The TYP. value is a reference value when V_{DD0} = V_{DD1} = V_{DD2} = 1.5 V, and T_A = 25°C.

2. The current consumed by the EV_{DD} system (output buffer and pull-up resistor) and the operating currents of A/D converters 0 to 2, the operational amplifier, and the comparator are not included.

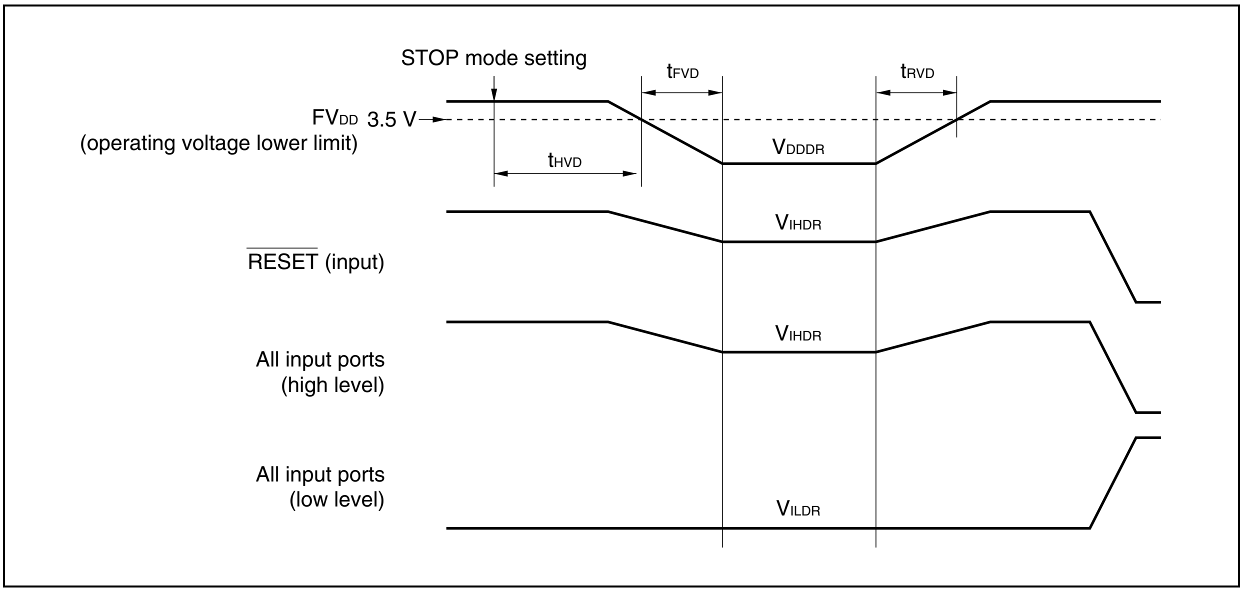
28.2.6 Data retention characteristics

STOP mode (T_A = –40 to +85°C, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode	**Note**		5.5	V
Data retention current	I_{DDDR}	V_{DD0} = V_{DD1} = V_{DD2} = V_{DDDR}		40	800	μA
Supply voltage rise time	t_{RVD}		1			μs
Supply voltage fall time	t_{FVD}		1			μs
Supply voltage retention time (from STOP mode setting)	t_{HVD}		0			ms
Data retention input voltage, high	V_{IHDR}	All input ports	$0.9V_{DDDR}$		V_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	All input ports	EV_{SS}		$0.1V_{DDDR}$	V

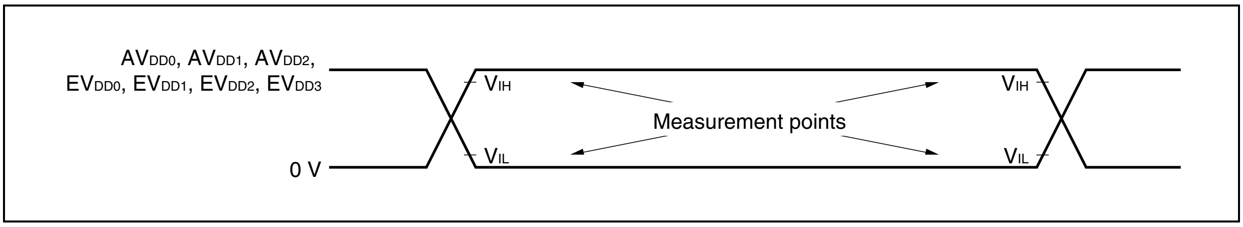
Note When the low-voltage detector (LVI) reset mode is not used (LVIM.LVIMD bit = 0): POC detection voltage (V_{POC0})

When the low-voltage detector (LVI) reset mode is used (LVIM.LVIMD bit = 1): LVI detection voltage (V_{LVI0}/V_{LVI1})

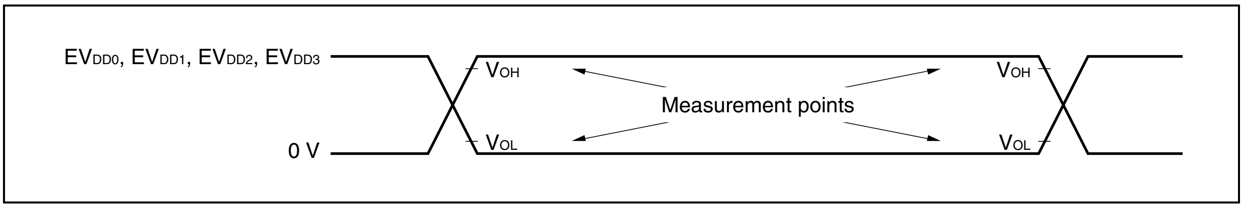

STOP mode setting
tFVD
tRVD
FVDD 3.5 V
(operating voltage lower limit)
tHVD
VDDDR
VIHDR
RESET (input)
VIHDR
All input ports (high level)
All input ports (low level)
VILDR

28.2.7 AC characteristics

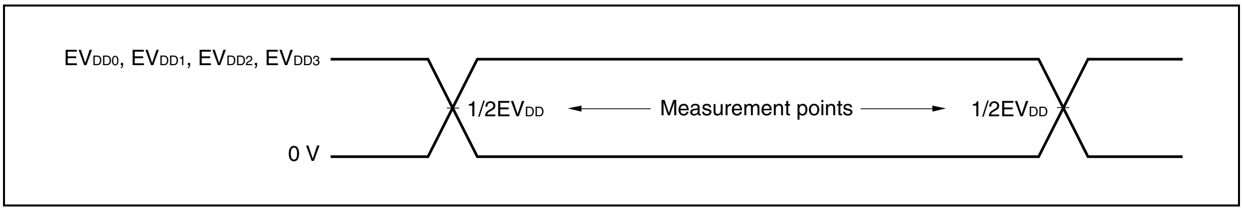
AC test input measurement points (external bus, pins other than CSIF0 to CSIF2)



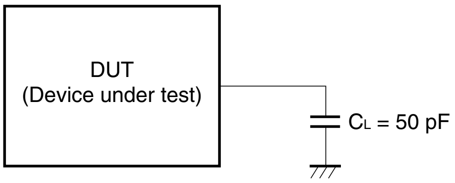
AC test output measurement points (external bus, pins other than CSIF0 to CSIF2)



AC test I/O measurement points (external bus, CSIF0 to CSIF2 pins)



Load conditions



Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

(1) Output signal timing

Caution There are specifications for EV_{DD} = 3.0 to 4.0 V and EV_{DD} = 4.0 to 5.5 V.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 4.0 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF) (1/2)

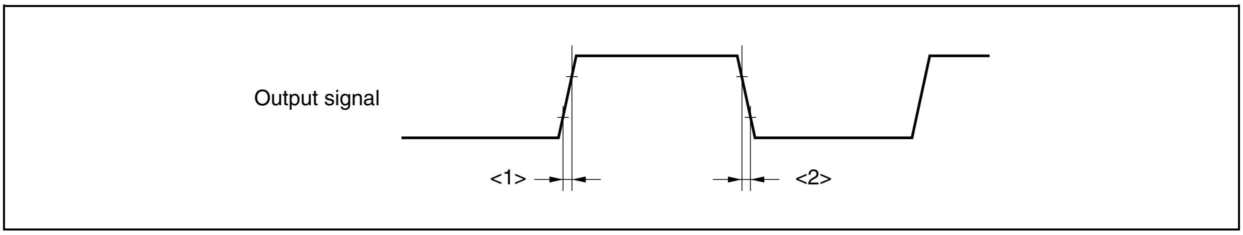
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	t_{OR}	<1>	P07, TRCCLK		10	ns
			Note		16	ns
			Other than above		30	ns
Output fall time	t_{OF}	<2>	P07, TRCCLK		10	ns
			Note		16	ns
			Other than above		30	ns

Note PDL0 to PDL15, DDO, TRCDATA0 to TRCDATA3, TRCEND

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 4.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF) (2/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	t_{OR}	<1>	P07, TRCCLK		5	ns
			Note		8	ns
			Other than above		15	ns
Output fall time	t_{OF}	<2>	P07, TRCCLK		5	ns
			Note		8	ns
			Other than above		15	ns

Note PDL0 to PDL15, DDO, TRCDATA0 to TRCDATA3, TRCEND



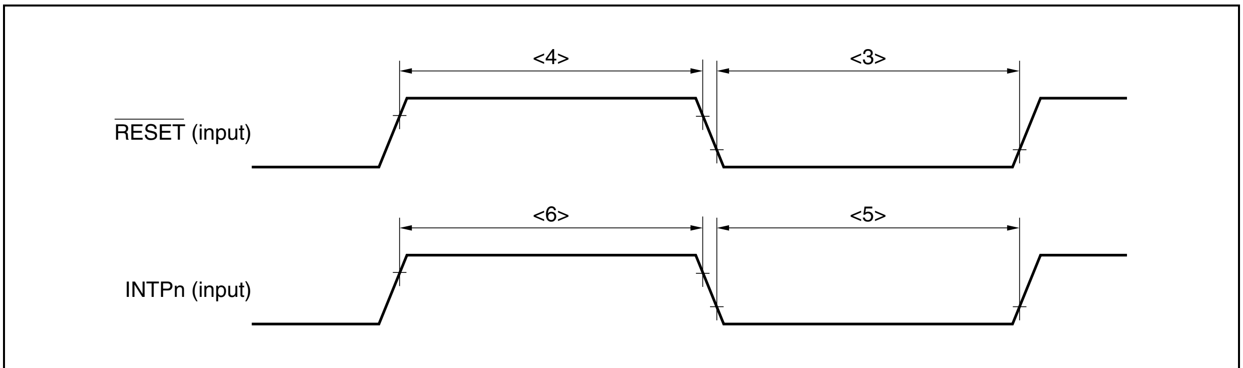
(2) Reset, external interrupt timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V,
FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<3>	Power is on, STOP mode is released	500 + T_{os}		ns
			Other than above	500		ns
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	<4>		500		ns
INTPn low-level width	t_{WITL}	<5>	n = 00 to 19 (analog noise elimination)	500		ns
			n = 00 to 02, 17 to 19 (digital noise elimination)	$4T_{smp}$		ns
INTPn high-level width	t_{WITH}	<6>	n = 00 to 19 (analog noise elimination)	500		ns
			n = 00 to 02, 17 to 19 (digital noise elimination)	$4T_{smp}$		ns

Remark T_{os}: Oscillation stabilization time
T_{smp}: Noise elimination sampling clock cycle (set by INTNFCn register)

Reset/Interrupt

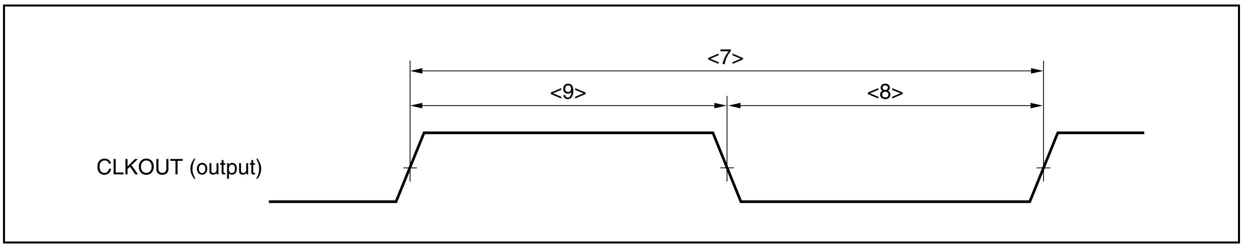


Remark n = 00 to 19

(3) CLKOUT output timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V,
FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<7>		31.25 ns	3.2 μs	
Low-level width	t_{WKH}	<8>		t_{CYK}/2 – 6.2		ns
High-level width	t_{WKL}	<9>		t_{CYK}/2 – 6.2		ns



(4) Bus timing

(a) Read cycle (CLKOUT asynchronous)

Cautions 1. There are specifications for EV_{DD} = 3.0 to 3.6 V and EV_{DD} = 4.0 to 5.5 V.

2. Set T in accordance with the following condition.

40 ns ≤ T

3. Be sure to insert the address setup waits and address hold waits.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 3.6 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V,
C_L = 50 pF) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from address to ASTB↓	t_{DAST2}	<10>		(0.5 + wAS) T – 12		ns
ASTB high-level width	t_{WSTH}	<11>		(1 + wAS + i) T – 12		ns
Address hold time from ASTB↓	t_{HSTA}	<12>		(0.5 + wAH) T – 18		ns
Address hold time from $\overline{RD}$↑	t_{HRDA2}	<13>		(1 + i) T – 11		ns
Delay time from address to $\overline{RD}$↓	t_{DARD2}	<14>		(1 + wAS + wAH) T – 13		ns
Delay time from $\overline{RD}$↓ to address float	t_{FRDA}	<15>			5	ns
Data input setup time from address	t_{DAID2}	<16>			(2 + wD + w + wAS + wAH) T – 39	ns
Data input setup time from ASTB↓	t_{DSTID}	<17>			(1.5 + wD + w + wAH) T – 38	ns
Data input setup time from $\overline{RD}$↓	t_{DRDID2}	<18>			(1 + wD + w) T – 36	ns
Delay time from ASTB↓ to $\overline{RD}$↓	t_{DSTRD3}	<19>		(0.5 + wAH) T – 5		ns
Data input hold time (from $\overline{RD}$↑)	t_{HRDID2}	<20>		0		ns
Delay time from $\overline{RD}$↑ to bus output	t_{DRDOD2}	<21>		(1 + i) T – 10		ns
Delay time from $\overline{RD}$↑ to ASTB↑	t_{DRDST}	<22>		0.5T – 5		ns
$\overline{RD}$ low-level width	t_{WRDL2}	<23>		(1 + wD + w) T – 10		ns
$\overline{RD}$ high-level width	t_{WRDH2}	<24>		(2 + i + wAS + wAH) T – 12		ns
High-level hold time from $\overline{RD}$↑ to $\overline{WRn}$	t_{HRDWR2}	<25>		(2 + i + wAS + wAH) T – 17		ns
$\overline{WAIT}$ setup time (to address)	t_{DAWT2}	<26>			(1.5 + wD + w + wAS + wAH) T – 43	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT2}	<27>		(1.5 + wD + w + wAS + wAH) T		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{DSTWT}	<28>			(1 + wD + w + wAH) T – 42	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT}	<29>		(1 + wD + w + wAH) T		ns
$\overline{WAIT}$ setup time (to $\overline{RD}$↓)	t_{DRDWT2}	<30>			(0.5 + wD + w) T – 40	ns
$\overline{WAIT}$ hold time (from $\overline{RD}$↓)	t_{HRDWT2}	<31>		(0.5 + wD + w) T		ns

Remarks 1. wAS: Number of address setup waits by the AWC register

wAH: Number of address hold waits by the AWC register

wD: Number of data waits by the DWC0 register

w: Number of external waits by the $\overline{WAIT}$ pin

2. T = 1/f_{CPU} (f_{CPU}: CPU clock frequency)

3. n = 0, 1

4. i: Number of idle states

Cautions 1. There are specifications for EV_{DD} = 3.0 to 3.6 V and EV_{DD} = 4.0 to 5.5 V.

2. Set T in accordance with the following condition.

40 ns ≤ T

3. Be sure to insert the address setup waits and address hold waits.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 4.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V,
C_L = 50 pF) (2/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from address to ASTB↓	t_{DAST2}	<10>		$(0.5 + w_{AS}) T - 12$		ns
ASTB high-level width	t_{WSTH}	<11>		$(1 + w_{AS} + i) T - 10$		ns
Address hold time from ASTB↓	t_{HSTA}	<12>		$(0.5 + w_{AH}) T - 10$		ns
Address hold time from $\overline{RD}$↑	t_{HRDA2}	<13>		$(1 + i) T - 10$		ns
Delay time from address to $\overline{RD}$↓	t_{DARD2}	<14>		$(1 + w_{AS} + w_{AH}) T - 12$		ns
Delay time from $\overline{RD}$↓ to address float	t_{FRDA}	<15>			7	ns
Data input setup time from address	t_{DAID2}	<16>			$(2 + w_D + w + w_{AS} + w_{AH}) T - 33$	ns
Data input setup time from ASTB↓	t_{DSTID}	<17>			$(1.5 + w_D + w + w_{AH}) T - 27$	ns
Data input setup time from $\overline{RD}$↓	t_{DRDID2}	<18>			$(1 + w_D + w) T - 25$	ns
Delay time from ASTB↓ to $\overline{RD}$↓	t_{DSTRD3}	<19>		$(0.5 + w_{AH}) T - 5$		ns
Data input hold time (from $\overline{RD}$↑)	t_{HRDID2}	<20>		0		ns
Delay time from $\overline{RD}$↑ to bus output	t_{DRDOD2}	<21>		$(1 + i) T - 5$		ns
Delay time from $\overline{RD}$↑ to ASTB↑	t_{DRDST}	<22>		$0.5T - 5$		ns
$\overline{RD}$ low-level width	t_{WRDL2}	<23>		$(1 + w_D + w) T - 10$		ns
$\overline{RD}$ high-level width	t_{WRDH2}	<24>		$(2 + i + w_{AS} + w_{AH}) T - 10$		ns
High-level hold time from $\overline{RD}$↑ to $\overline{WRn}$	t_{HRDWR2}	<25>		$(2 + i + w_{AS} + w_{AH}) T - 13$		ns
$\overline{WAIT}$ setup time (to address)	t_{DAWT2}	<26>			$(1.5 + w_D + w + w_{AS} + w_{AH}) T - 36$	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT2}	<27>		$(1.5 + w_D + w + w_{AS} + w_{AH}) T$		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{DSTWT}	<28>			$(1 + w_D + w + w_{AH}) T - 30$	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT}	<29>		$(1 + w_D + w + w_{AH}) T$		ns
$\overline{WAIT}$ setup time (to $\overline{RD}$↓)	t_{DRDWT2}	<30>			$(0.5 + w_D + w) T - 29$	ns
$\overline{WAIT}$ hold time (from $\overline{RD}$↓)	t_{HRDWT2}	<31>		$(0.5 + w_D + w) T$		ns

Remarks 1. w_{AS}: Number of address setup waits by the AWC register

w_{AH}: Number of address hold waits by the AWC register

w_D: Number of data waits by the DWC0 register

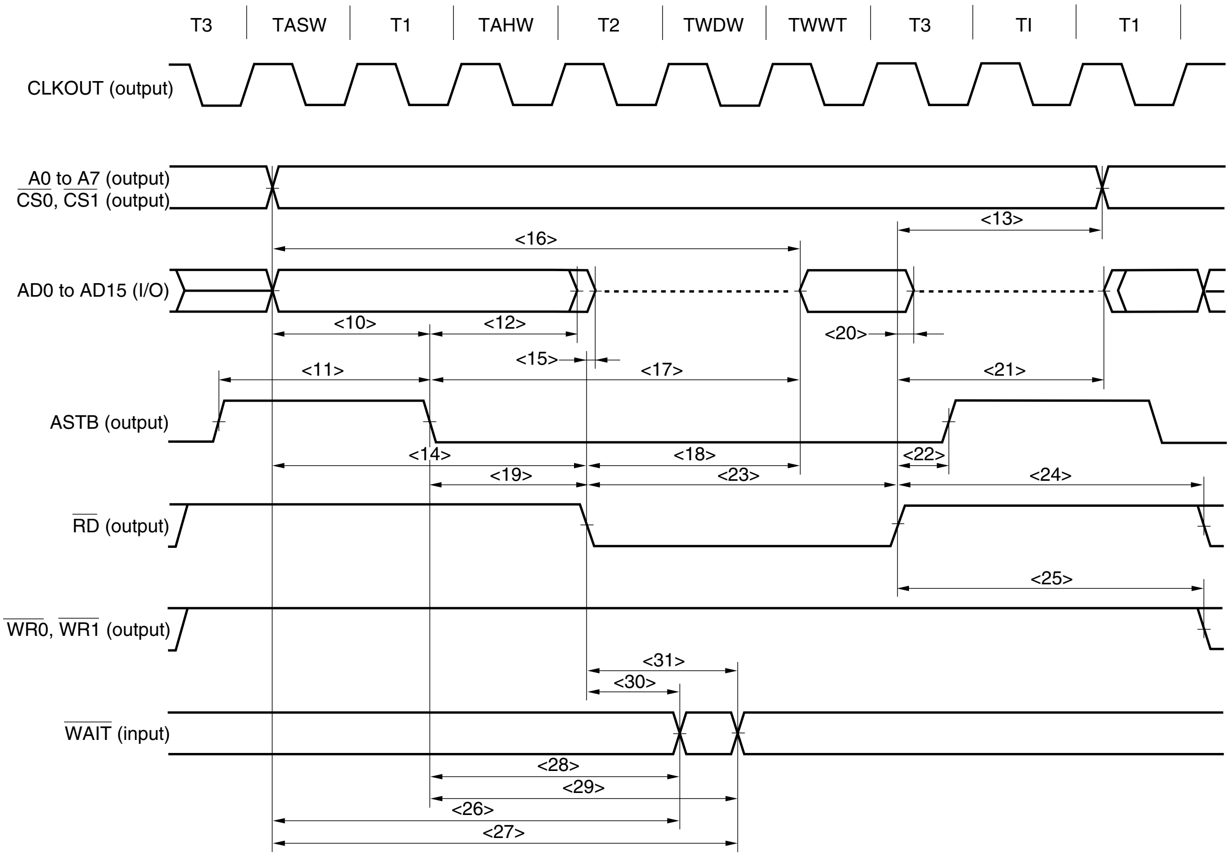
w: Number of external waits by the $\overline{WAIT}$ pin

2. T = 1/f_{CPU} (f_{CPU}: CPU clock frequency)

3. n = 0, 1

4. i: Number of idle states

Read cycle (CLKOUT asynchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, number of waits by $\overline{\text{WAIT}}$ pin is 1 (when an active level (low level) is input for one cycle during the determined wait period), and number of idle states is 1.

(b) Read cycle (CLKOUT synchronous)

Cautions 1. There are specifications for EV_{DD} = 3.0 to 3.6 V and EV_{DD} = 4.0 to 5.5 V.

2. Be sure to insert the address setup waits and address hold waits.

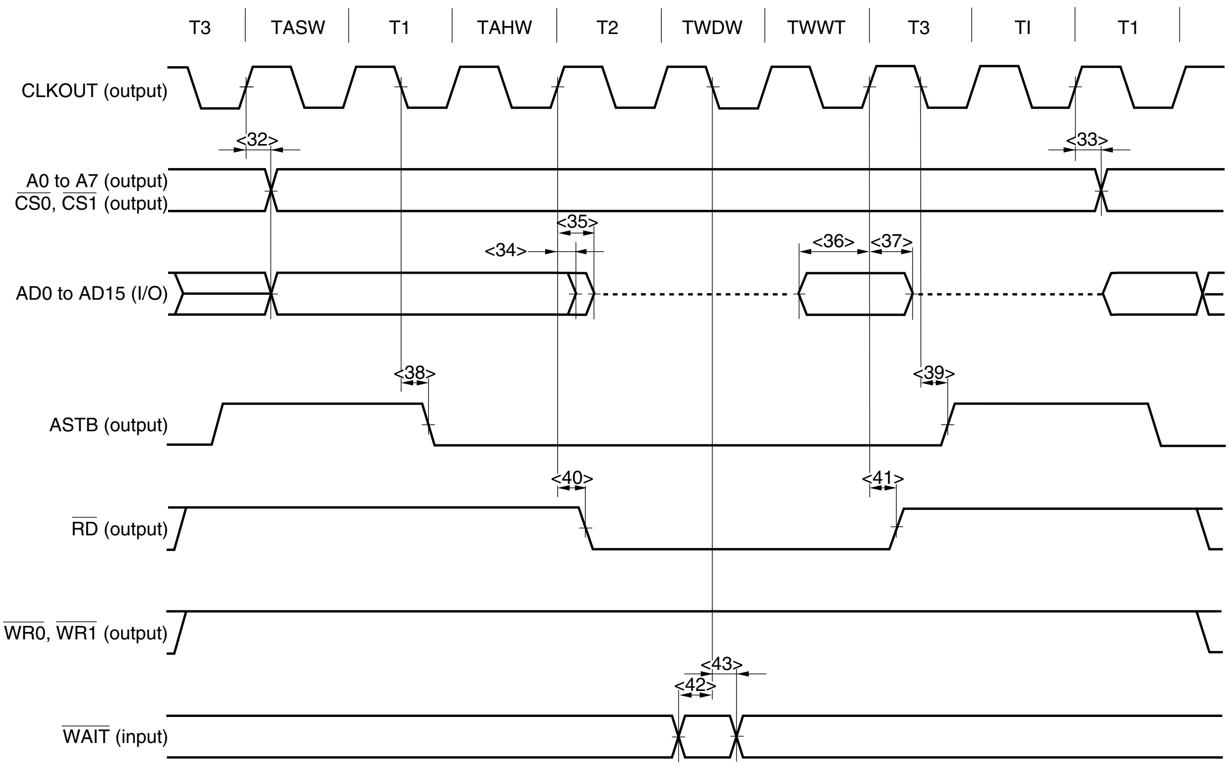
(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 3.6 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V,
C_L = 50 pF) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA2}	<32>			19	ns
Address hold time from CLKOUT↑	t_{HKA2}	<33>		0		ns
Address hold time from CLKOUT↑	t_{HKA3}	<34>		–1		ns
Delay time from CLKOUT↑ to address float	t_{FKA}	<35>			12	ns
Data input setup time (to CLKOUT↑)	t_{SIDK2}	<36>		21		ns
Data input hold time (from CLKOUT↑)	t_{HKID2}	<37>		0		ns
Delay time from CLKOUT↓ to ASTB↓	t_{DKST3}	<38>		0	17	ns
Delay time from CLKOUT↓ to ASTB↑	t_{DKST4}	<39>		0	16	ns
Delay time from CLKOUT↑ to $\overline{\text{RD}}$↓	t_{DKRD3}	<40>		0	16	ns
Delay time from CLKOUT↑ to $\overline{\text{RD}}$↑	t_{DKRD4}	<41>		0	15	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↓)	t_{SWTK2}	<42>		25		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↓)	t_{HKWT2}	<43>		0		ns

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 4.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V,
C_L = 50 pF) (2/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA2}	<32>			16	ns
Address hold time from CLKOUT↑	t_{HKA2}	<33>		0		ns
Address hold time from CLKOUT↑	t_{HKA3}	<34>		–1		ns
Delay time from CLKOUT↑ to address float	t_{FKA}	<35>			12	ns
Data input setup time (to CLKOUT↑)	t_{SIDK2}	<36>		18		ns
Data input hold time (from CLKOUT↑)	t_{HKID2}	<37>		0		ns
Delay time from CLKOUT↓ to ASTB↓	t_{DKST3}	<38>		0	12	ns
Delay time from CLKOUT↓ to ASTB↑	t_{DKST4}	<39>		0	13	ns
Delay time from CLKOUT↑ to $\overline{\text{RD}}$↓	t_{DKRD3}	<40>		0	12	ns
Delay time from CLKOUT↑ to $\overline{\text{RD}}$↑	t_{DKRD4}	<41>		0	12	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↓)	t_{SWTK2}	<42>		21		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↓)	t_{HKWT2}	<43>		0		ns

Read cycle (CLKOUT synchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, number of waits by $\overline{\text{WAIT}}$ pin is 1 (when an active level (low level) is input for one cycle during the determined wait period), and number of idle states is 1.

(c) Write cycle (CLKOUT asynchronous)

Cautions 1. There are specifications for EV_{DD} = 3.0 to 3.6 V and EV_{DD} = 4.0 to 5.5 V.

2. Set T in accordance with the following condition.

40 ns ≤ T

3. Be sure to insert the address setup waits and address hold waits.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 3.6 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V,
C_L = 50 pF) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from address to ASTB↓	t_{DAST2}	<10>		(0.5 + WAS) T – 12		ns
ASTB high-level width	t_{WSTH}	<11>		(1 + WAS + i) T – 12		ns
Address hold time from ASTB↓	t_{HSTA}	<12>		(0.5 + WAH) T – 18		ns
Address hold time from $\overline{WRn}$↑	t_{HWRA2}	<44>		T – 16		ns
Delay time from address to $\overline{WRn}$↓	t_{DAWR2}	<45>		(1 + WAS + WAH) T – 12		ns
Delay time from $\overline{WRn}$↓ to data output	t_{DWROD3}	<46>			5	ns
Delay time from ASTB↓ to $\overline{WRn}$↓	t_{DSTWR3}	<47>		(0.5 + WAH) T – 9		ns
Delay time from data output to $\overline{WRn}$↑	t_{DODWR2}	<48>		(1 + WD + w) T – 10		ns
Data output hold time from $\overline{WRn}$↑	t_{HWROD2}	<49>		T – 15		ns
Delay time from $\overline{WRn}$↑ to ASTB↑	t_{DWRST}	<50>		0.5T – 7		ns
$\overline{WRn}$ low-level width	t_{WWRL2}	<51>		(1 + WD + w) T – 12		ns
$\overline{WRn}$ high-level width	t_{WWRH2}	<52>		(2 + WAS + WAH) T – 10		ns
High-level hold time from $\overline{WRn}$↑ to $\overline{RD}$	t_{HWRRD2}	<53>		(2 + WAS + WAH) T – 17		ns
$\overline{WAIT}$ setup time (to address)	t_{DAWT2}	<26>			(1.5 + WD + w + WAS + WAH) T – 43	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT2}	<27>		(1.5 + WD + w + WAS + WAH) T		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{DSTWT}	<28>			(1 + WD + w + WAH) T – 42	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT}	<29>		(1 + WD + w + WAH) T		ns
$\overline{WAIT}$ setup time (to $\overline{WRn}$↓)	t_{DWRWT2}	<54>			(0.5 + WD + w) T – 49	ns
$\overline{WAIT}$ hold time (from $\overline{WRn}$↓)	t_{HWRWT2}	<55>		(0.5 + WD + w) T		ns

Remarks 1. WAS: Number of address setup waits by the AWC register
WAH: Number of address hold waits by the AWC register
WD: Number of data waits by the DWC0 register
w: Number of external waits by the $\overline{WAIT}$ pin

2. T = $1/f_{CPU}$ (f_{CPU}: CPU operating clock frequency)

3. n = 0, 1

4. i: Number of idle states

Cautions 1. There are specifications for EV_{DD} = 3.0 to 3.6 V and EV_{DD} = 4.0 to 5.5 V.

2. Set T in accordance with the following condition.

40 ns ≤ T

3. Be sure to insert the address setup waits and address hold waits.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 4.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V,
C_L = 50 pF) (2/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from address to ASTB↓	t_{DAST2}	<10>		(0.5 + w_{AS}) T – 12		ns
ASTB high-level width	t_{WSTH}	<11>		(1 + w_{AS} + i) T – 10		ns
Address hold time from ASTB↓	t_{HSTA}	<12>		(0.5 + w_{AH}) T – 10		ns
Address hold time from $\overline{WRn}$↑	t_{HWRA2}	<44>		T – 13		ns
Delay time from address to $\overline{WRn}$↓	t_{DAWR2}	<45>		(1 + w_{AS} + w_{AH}) T – 10		ns
Delay time from $\overline{WRn}$↓ to data output	t_{DWROD3}	<46>			7	ns
Delay time from ASTB↓ to $\overline{WRn}$↓	t_{DSTWR3}	<47>		(0.5 + w_{AH}) T – 9		ns
Delay time from data output to $\overline{WRn}$↑	t_{DODWR2}	<48>		(1 + w_D + w) T – 10		ns
Data output hold time from $\overline{WRn}$↑	t_{HWROD2}	<49>		T – 12		ns
Delay time from $\overline{WRn}$↑ to ASTB↑	t_{DWRST}	<50>		0.5T – 7		ns
$\overline{WRn}$ low-level width	t_{WWRL2}	<51>		(1 + w_D + w) T – 10		ns
$\overline{WRn}$ high-level width	t_{WWRH2}	<52>		(2 + w_{AS} + w_{AH}) T – 10		ns
High-level hold time from $\overline{WRn}$↑ to $\overline{RD}$	t_{HWRRD2}	<53>		(2 + w_{AS} + w_{AH}) T – 14		ns
$\overline{WAIT}$ setup time (to address)	t_{DAWT2}	<26>			(1.5 + w_D + w + w_{AS} + w_{AH}) T – 36	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT2}	<27>		(1.5 + w_D + w + w_{AS} + w_{AH}) T		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{DSTWT}	<28>			(1 + w_D + w + w_{AH}) T – 30	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT}	<29>		(1 + w_D + w + w_{AH}) T		ns
$\overline{WAIT}$ setup time (to $\overline{WRn}$↓)	t_{DWRWT2}	<54>			(0.5 + w_D + w) T – 39	ns
$\overline{WAIT}$ hold time (from $\overline{WRn}$↓)	t_{HWRWT2}	<55>		(0.5 + w_D + w) T		ns

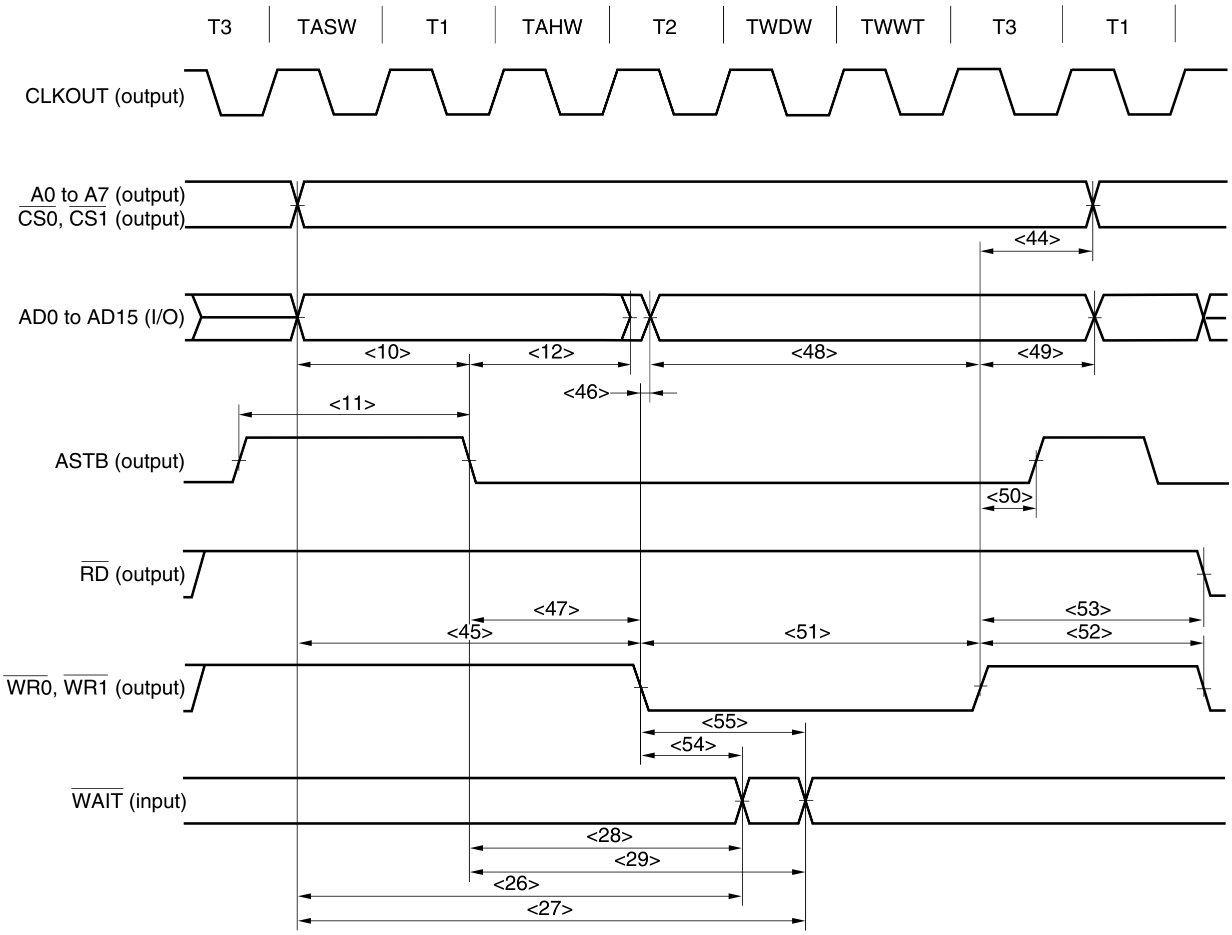
Remarks 1. w_{AS}: Number of address setup waits by the AWC register
w_{AH}: Number of address hold waits by the AWC register
w_D: Number of data waits by the DWC0 register
w: Number of external waits by the $\overline{WAIT}$ pin

2. T = 1/f_{CPU} (f_{CPU}: CPU operating clock frequency)

3. n = 0, 1

4. i: Number of idle states

Write cycle (CLKOUT asynchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, and number of waits by WAIT pin is 1 (when an active level (low level) is input for one cycle during the determined wait period).

(d) Write cycle (CLKOUT synchronous)

Cautions 1. There are specifications for EV_{DD} = 3.0 to 3.6 V and EV_{DD} = 4.0 to 5.5 V.

2. Be sure to insert the address setup waits and address hold waits.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 3.6 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V,
C_L = 50 pF) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA2}	<32>			19	ns
Address hold time from CLKOUT↑	t_{HKA2}	<33>		0		ns
Address hold time from CLKOUT↑	t_{HKA3}	<34>		–1		ns
Delay time from CLKOUT↓ to ASTB↓	t_{DKST3}	<38>		0	17	ns
Delay time from CLKOUT↓ to ASTB↑	t_{DKST4}	<39>		0	16	ns
Delay time from CLKOUT↑ to data output	t_{DKOD3}	<56>			13	ns
Data output hold time from CLKOUT↑	t_{HKOD2}	<57>		0		ns
Delay time from CLKOUT↑ to $\overline{\text{WRn}}$↓	t_{DKWR3}	<58>		0	25	ns
Delay time from CLKOUT↑ to $\overline{\text{WRn}}$↑	t_{DKWR4}	<59>		0	23	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↓)	t_{SWTK2}	<42>		25		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↓)	t_{HKWT2}	<43>		0		ns

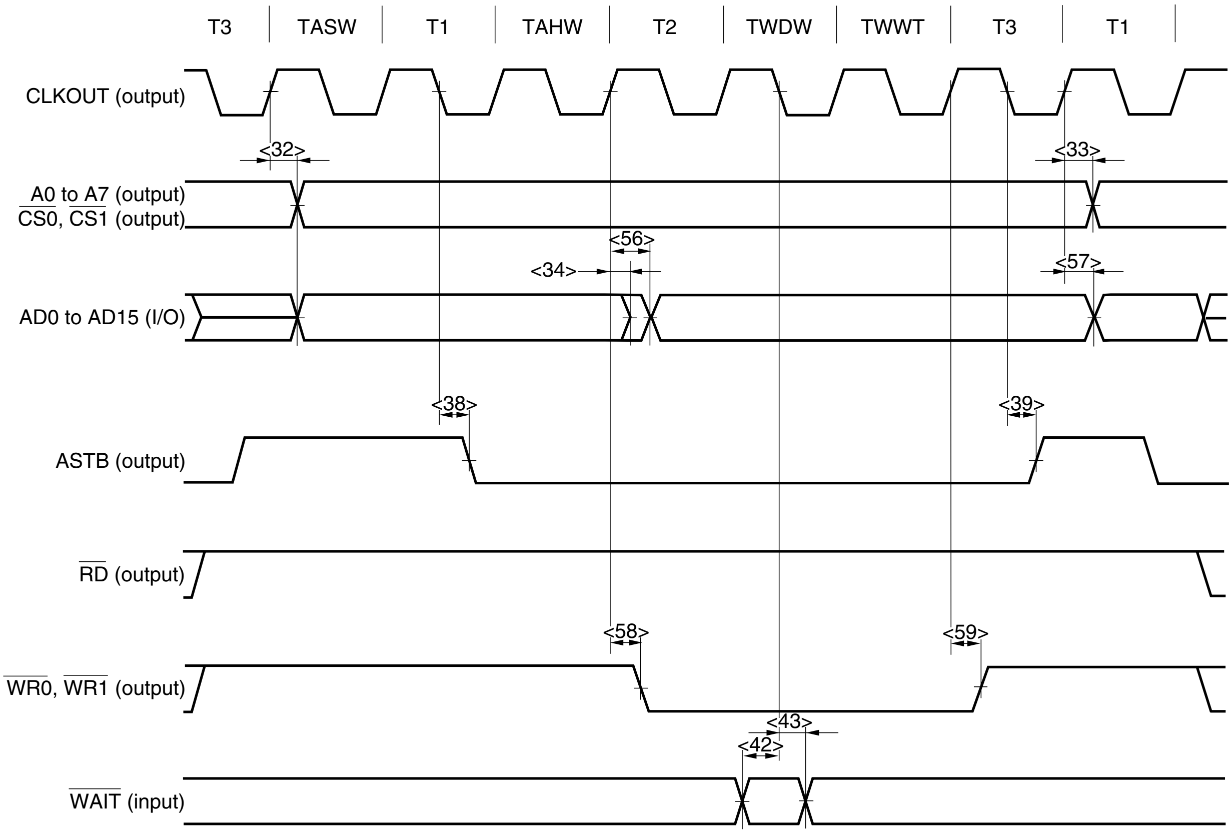
Remark n = 0, 1

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 4.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V,
C_L = 50 pF) (2/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA2}	<32>			16	ns
Address hold time from CLKOUT↑	t_{HKA2}	<33>		0		ns
Address hold time from CLKOUT↑	t_{HKA3}	<34>		–1		ns
Delay time from CLKOUT↓ to ASTB↓	t_{DKST3}	<38>		0	12	ns
Delay time from CLKOUT↓ to ASTB↑	t_{DKST4}	<39>		0	13	ns
Delay time from CLKOUT↑ to data output	t_{DKOD3}	<56>			12	ns
Data output hold time from CLKOUT↑	t_{HKOD2}	<57>		0		ns
Delay time from CLKOUT↑ to $\overline{\text{WRn}}$↓	t_{DKWR3}	<58>		0	18	ns
Delay time from CLKOUT↑ to $\overline{\text{WRn}}$↑	t_{DKWR4}	<59>		0	18	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↓)	t_{SWTK2}	<42>		21		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↓)	t_{HKWT2}	<43>		0		ns

Remark n = 0, 1

Write cycle (CLKOUT synchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, and number of waits by WAIT pin is 1 (when an active level (low level) is input for one cycle during the determined wait period).

(3) Timer timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIn high-level width[Notes 1, 2]	t_{WTIHn}	<60>	n = B00 to B03, B10 to B13	12T + 10		ns
			n = A20, A21	$3T_{smp1}$ + 10		ns
TIn low-level width[Notes 1, 2]	t_{WTILn}	<61>	n = B00 to B03, B10 to B13	12T + 10		ns
			n = A20, A21	$3T_{smp1}$ + 10		ns
EVTBm high-level width[Note 1]	t_{WEVBHm}	<62>	m = 0, 1	12T + 10		ns
EVTBm low-level width[Note 1]	t_{WEVBLm}	<63>	m = 0, 1	12T + 10		ns
TRGBm high-level width[Note 1]	t_{WTRHm}	<64>	m = 0, 1	12T + 10		ns
TRGBm low-level width[Note 1]	t_{WTRLm}	<65>	m = 0, 1	12T + 10		ns
TENCm0/TENCm1 high-level width[Note 3]	t_{WENCHm}	<66>	m = 0, 1	$3T_{smp2}$ + 10		ns
TENCm0/TENCm1 low-level width[Note 3]	t_{WENCLm}	<67>	m = 0, 1	$3T_{smp2}$ + 10		ns
TECRm high-level width[Note 3]	t_{WCRHm}	<68>	m = 0, 1	$3T_{smp2}$ + 10		ns
TECRm low-level width[Note 3]	t_{WCRLm}	<69>	m = 0, 1	$3T_{smp2}$ + 10		ns
TITk0/TITk1 high-level width[Note 3]	t_{WTITHm}	<70>	k = 0 to 3	$3T_{smp2}$ + 10		ns
TITk0/TITk1 low-level width[Note 3]	t_{WTITLm}	<71>	k = 0 to 3	$3T_{smp2}$ + 10		ns
EVTTm high-level width[Note 3]	t_{WEVTHm}	<72>	m = 0, 1	$3T_{smp2}$ + 10		ns
EVTTm low-level width[Note 3]	t_{WEVTLm}	<73>	m = 0, 1	$3T_{smp2}$ + 10		ns
TENCm0/TENCm1 input time differential[Note 3]	t_{PHUDm}	<74>	m = 0, 1	$3T_{smp2}$ + 10		ns

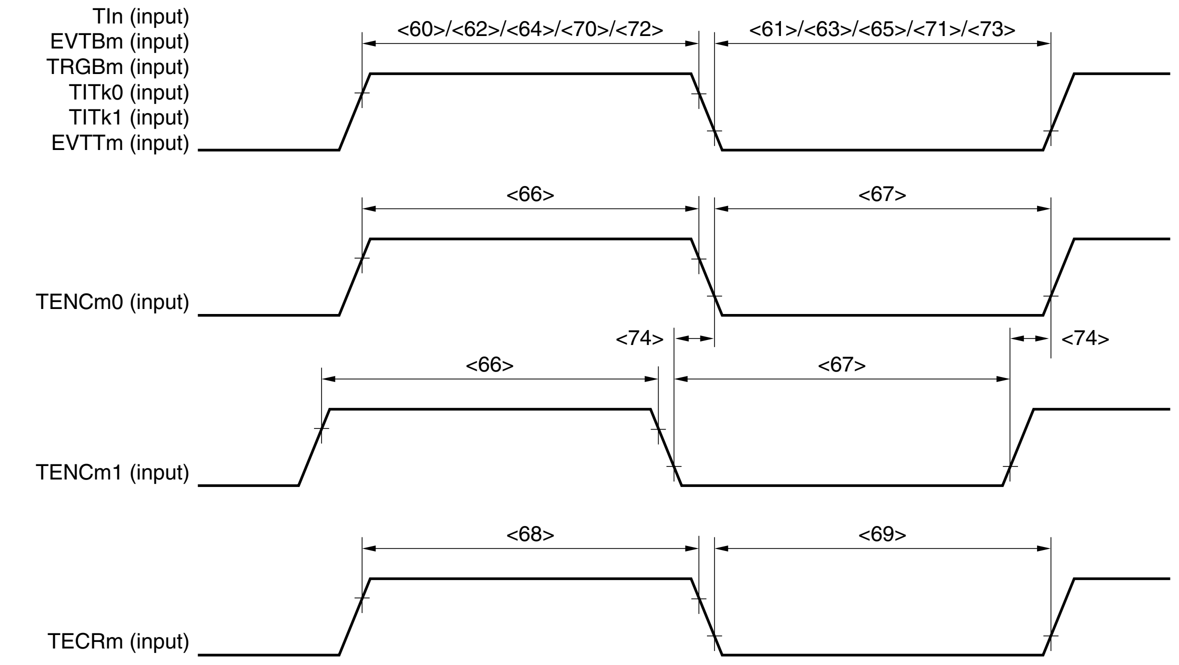
Notes 1. T = 1/fxx

2. T_{smp1}: Noise elimination sampling clock cycle (set by TANFC2 register)

3. T_{smp2}: Noise elimination sampling clock cycle (set by TTNFC0 and TTNFC3 registers)

Remark The above specification shows a pulse width that is accurately detected as a valid edge. Even if a pulse narrower than the above specification is input, therefore, it may be detected as a valid edge.

Timer Input Timing



Remark n = A20, A21, B00 to B03, B10 to B13
k = 0 to 3
m = 0, 1

(4) UARTA timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transmission rate				1.25	Mbps

(5) UARTB timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transmission rate				5.00	Mbps

(6) CSIF timing

(a) Master mode

Caution There are specifications for EV_{DD} = 3.0 to 4.0 V and EV_{DD} = 4.0 to 5.5 V.

**(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 4.0 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF) (1/2)**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKFn}}$ cycle	t_{KCYM}	<75>		125		ns
$\overline{\text{SCKFn}}$ high-/low-level width	t_{KWHM}, t_{KWLM}	<76>		$t_{KCYM}/2 - 15$		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↑)	t_{SSIM}	<77>		40		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↓)				40		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSIM}	<78>		40		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↓)				40		ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↓)	t_{DSOM}	<79>			30	ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↑)					30	ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSOM}	<80>		$t_{KCYM}/2 - 10$		ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↓)				$t_{KCYM}/2 - 10$		ns

Remark n = 0 to 2

**(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 4.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF) (2/2)**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKFn}}$ cycle	t_{KCYM}	<75>		125		ns
$\overline{\text{SCKFn}}$ high-/low-level width	t_{KWHM}, t_{KWLM}	<76>		$t_{KCYM}/2 - 10$		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↑)	t_{SSIM}	<77>		30		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↓)				30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSIM}	<78>		30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↓)				30		ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↓)	t_{DSOM}	<79>			30	ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↑)					30	ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSOM}	<80>		$t_{KCYM}/2 - 10$		ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↓)				$t_{KCYM}/2 - 10$		ns

Remark n = 0 to 2

(b) Slave mode

Caution There are specifications for EV_{DD} = 3.0 to 4.0 V and EV_{DD} = 4.0 to 5.5 V.

**(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 4.0 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF) (1/2)**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKFn}}$ cycle	t_{KCYS}	<81>		125		ns
$\overline{\text{SCKFn}}$ high-/low-level width	t_{KWHS}, t_{KWLS}	<82>		$t_{KCYS}/2 - 10$		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↑)	t_{SSIS}	<83>		30		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↓)				30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSIS}	<84>		30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↓)				30		ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↓)	t_{DSOS}	<85>			40	ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↑)					40	ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSOS}	<86>		$t_{KCYS}/2 - 10$		ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↓)				$t_{KCYS}/2 - 10$		ns

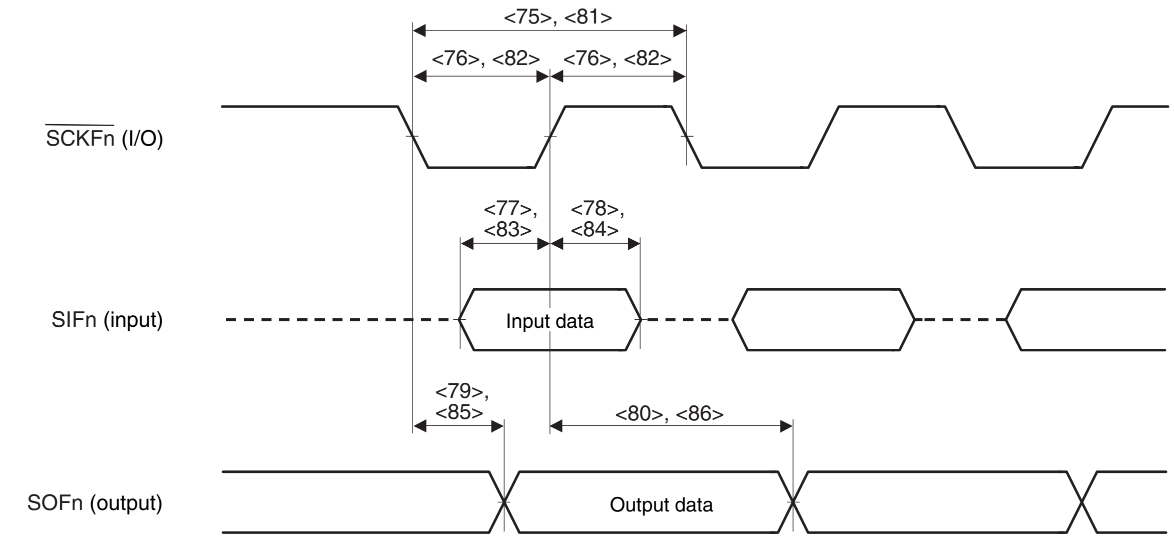
Remark n = 0 to 2

**(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 4.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF) (2/2)**

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKFn}}$ cycle	t_{KCYS}	<81>		125		ns
$\overline{\text{SCKFn}}$ high-/low-level width	t_{KWHS}, t_{KWLS}	<82>		$t_{KCYS}/2 - 10$		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↑)	t_{SSIS}	<83>		30		ns
SIFn setup time (to $\overline{\text{SCKFn}}$↓)				30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSIS}	<84>		30		ns
SIFn hold time (from $\overline{\text{SCKFn}}$↓)				30		ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↓)	t_{DSOS}	<85>			30	ns
SOFn output delay time (from $\overline{\text{SCKFn}}$↑)					30	ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↑)	t_{HSOS}	<86>		$t_{KCYS}/2 - 10$		ns
SOFn output hold time (from $\overline{\text{SCKFn}}$↓)				$t_{KCYS}/2 - 10$		ns

Remark n = 0 to 2

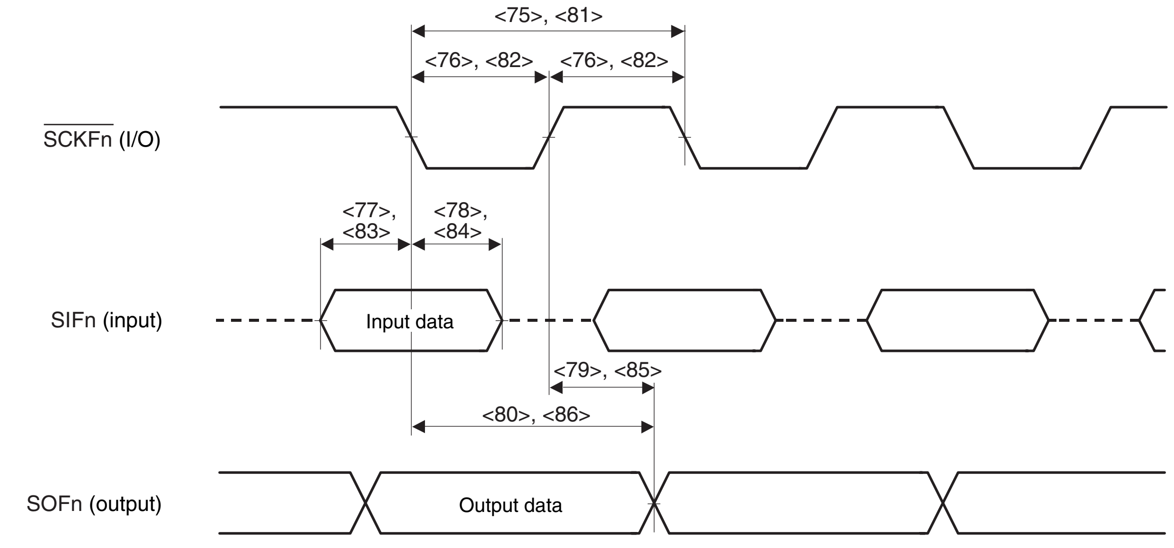
CSIF timing when CFnCKP and CFnDAP bits of CFnCTL1 register = 00



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

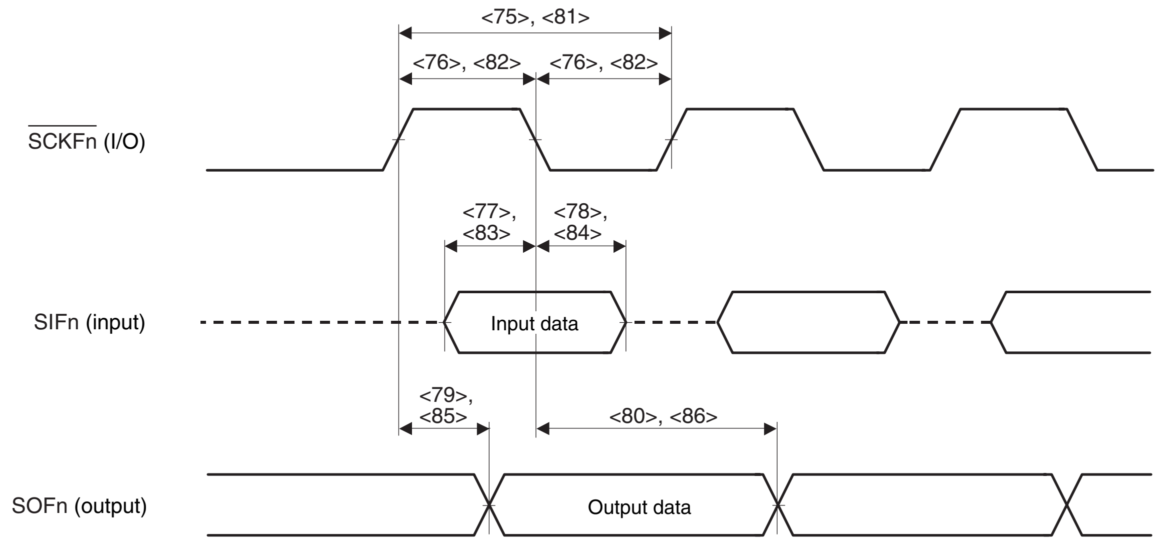
CSIF timing when CFnCKP and CFnDAP bits of CFnCTL1 register = 01



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

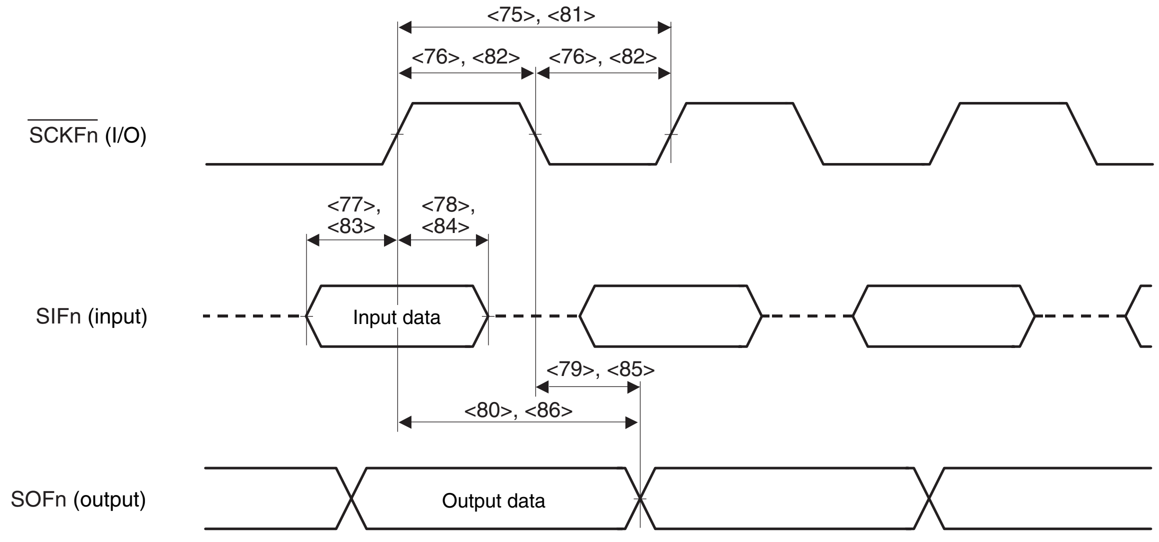
CSIF timing when CFnCKP and CFnDAP bits of CFnCTL1 register = 10



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

CSIF timing when CFnCKP and CFnDAP bits of CFnCTL1 register = 11



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

(7) I^{2}C bus timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter		Symbol		Standard Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL clock frequency		f_{CLK}	–	0	100	0	400	kHz
Bus free time (between stop condition and start condition)		t_{BUF}	<87>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<88>	4.0	–	0.6	–	μs
SCL clock low-level width		t_{LOW}	<89>	4.7	–	1.3	–	μs
SCL clock high-level width		t_{HIGH}	<90>	4.0	–	0.6	–	μs
Start/restart condition setup time		$t_{SU:STA}$	<91>	4.7	–	0.6	–	μs
Data hold time	CBUS-compatible master	$t_{HD:DAT}$	<92>	5.0	–	–	–	μs
	I^{2}C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<93>	250	–	100[Note 4]	–	ns
SDA, SCL signal rise time		t_R	<94>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA, SCL signal fall time		t_F	<95>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<96>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<97>	–	–	0	50	ns
Each bus line capacitive load		Cb	–	–	400	–	400	pF

Notes 1. The first clock pulse is generated after a hold time during the start condition.

2. The system must internally supply a hold time of at least 300 ns for the SDA signal (at $V_{IHmin.}$ of SCL signal) to fill the undefined area at the falling edge of SCL.

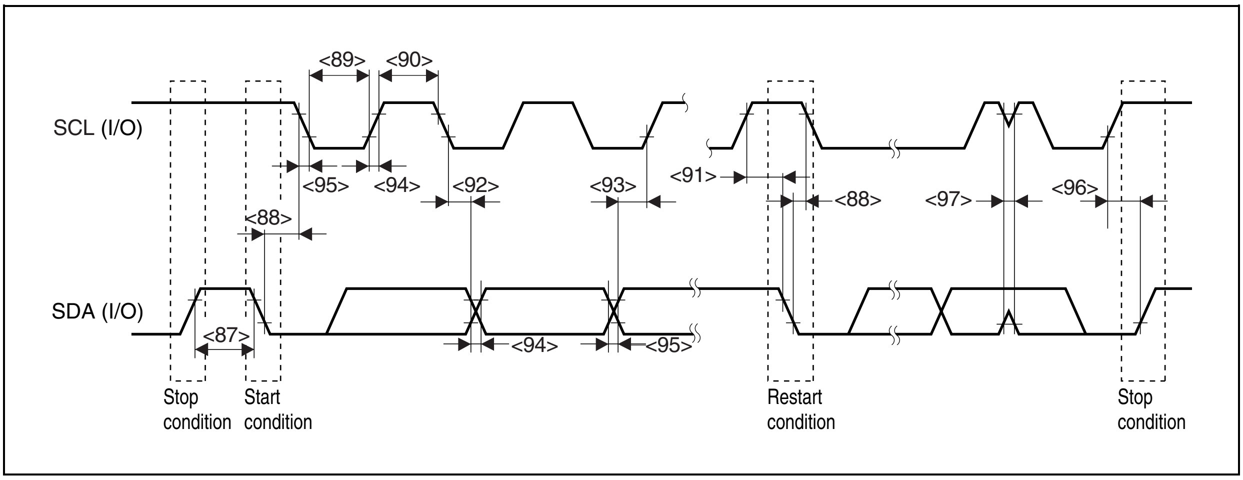
3. If the system does not extend the low hold time (t_{LOW}) of the SCL signal, the maximum data hold time ($t_{HD:DAT}$) must be satisfied.

4. The high-speed mode I^{2}C bus can be used in the standard mode I^{2}C bus system. In this case, make sure that the following conditions are satisfied.

- If system does not extend the low status hold time of the SCL signal
  $t_{SU:\ DAT} \geq 250$ ns
- If system extends the low status hold time of SCL signal
  Sends the next data bit to the SDA line before the SCL line is released ($t_{Rmax.}$ + $t_{SU:DAT}$ = 1000 + 250 = 1250 ns: standard mode I^{2}C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C bus timing



(8) High-impedance control timing

(T$_A$ = –40 to +85°C, V$_{DD0}$ = V$_{DD1}$ = V$_{DD2}$ = 1.35 to 1.65 V,

EV$_{DD0}$ = EV$_{DD1}$ = EV$_{DD2}$ = EV$_{DD3}$ = 3.0 to 5.5 V, FV$_{DD}$ = AV$_{DD0}$ = AV$_{DD1}$ = AV$_{DD2}$ = 4.0 to 5.5 V, UV$_{DD}$ = 3.0 to 3.6 V,

V$_{SS0}$ = V$_{SS1}$ = V$_{SS2}$ = EV$_{SS0}$ = EV$_{SS1}$ = EV$_{SS2}$ = EV$_{SS3}$ = EV$_{SS4}$ = AV$_{SS0}$ = AV$_{SS1}$ = AV$_{SS2}$ = 0 V, C$_L$ = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Oscillation stop → timer output high impedance	t_{CLM}	When clock monitor is operating		65	μs
Input to TOBnOFF, TOB01OFF → timer output high impedance	t_{HTQn}			300	ns
Input to TOTmOFF → timer output high impedance	t_{HTPm}			300	ns
Input to ANI00/ANI05 to ANI02/ANI07 → timer output high impedance	t_{ANI0}			10	μs
Input to ANI10/ANI15 to ANI12/ANI17 → timer output high impedance	t_{ANI1}			10	μs

Remark n = 0, 1
m = 2, 3

28.2.8 Characteristics of A/D converters 0, 1

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = AV_{REFP0} = AV_{REFP1} = 4.0 to 5.5 V,
UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			12	12	12	bit
Overall error[Note 1]					±10	LSB
Conversion time	t_{CONV}		2.00			μs
					8.00	μs
Zero scale error[Note 1]					±10	LSB
Full-scale error[Note 1]					±10	LSB
Integral linearity error[Note 1]					±4	LSB
Differential linearity error[Note 1]					±2.5	LSB
Analog reference voltage	AV_{REF}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current[Note 2]	AI_{DD}	During operation		4.5	7.5	mA
	AI_{DDS}	In STOP mode[Note 3]		3.5	17.5	μA

Notes 1. Excludes quantization error (±0.5 LSB).

2. This value is for only one A/D converter (A/D converter 0 or 1).

3. Stop A/D converters 0 and 1 (ADnSCM.ADnCE bit = 0) before setting STOP mode.

Remarks 1. LSB: Least Significant Bit

2. f_{AD01}: Base clock for A/D converters 0 and 1

3. n = 0, 1

28.2.9 Characteristics of A/D converter 2

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error Note 1					±4.0	LSB
Conversion time	t_{CONV}		3.00		10.00	μs
Zero scale error Note 1					±4.0	LSB
Full-scale error Note 1					±4.0	LSB
Integral linearity error Note 1					±4.0	LSB
Differential linearity error Note 1					±2.0	LSB
Analog reference voltage	AV_{REF}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current	AI_{DD}	During operation		3.5	7	mA
	AI_{DDS}	In STOP mode Note 2		1	10	μA

Notes 1. Excludes quantization error (±0.5LSB).

2. Stop the operation of A/D converter 2 (AD2M0.AD2CE bit = 0) before setting STOP mode.

Remark LSB: Least Significant Bit

28.2.10 Operational amplifier characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V, EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V, V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IO}				±9.0		mV
Input voltage range	V_I	Gain = 2.500		0.04AV_{DD}		0.36AV_{DD}	V
		Gain = 5.000		0.02AV_{DD}		0.18AV_{DD}	V
		Gain = 10.00		0.01AV_{DD}		0.085AV_{DD}	V
Slew rate[Note 1]	S_R			10	15		V/μs
Gain error		**Note 2**	Gain = 2.500 to 4.444		±1.0	±1.3	%
			Gain = 5.000 to 6.667		±1.0	±1.5	%
			Gain = 8.000, 10.00		±1.0	±1.7	%
		Note 3	Gain = 2.500 to 4.444		±1.0	±2.0	%
			Gain = 5.000 to 6.667		±1.0	±2.1	%
			Gain = 8.000, 10.00		±1.0	±2.2	%
Operating current[Note 4]	I_{OPDD}	During operation			1.8	2.6	mA
	A_{IDDS}	In STOP mode[Note 5]			1.0	10	μA

Notes 1. Inclination characteristic of 10% to 90% of output voltage

2. 4.5 V ≤ AV_{DD0} = AV_{DD1} ≤ 5.5 V

3. 4.0 V ≤ AV_{DD0} = AV_{DD1} < 4.5 V

4. Six operational amplifiers are provided in total. The value shows the operating current per operational amplifier.

5. Stop operational amplifier operation (OPnCTL0.OPn2EN to OPn0EN bits = 000) before setting STOP mode.

Remark Power supplies AV_{DD0} and AV_{DD1} are used for the operational amplifier.

28.2.11 Comparator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IO}			±3.0		mV
Input voltage range	V_I		AV_{SS}		AV_{DD}	V
Response time	t_{CR}	Input amplitude = 100 mV, at rising edge[Note 1]		1.0		μs
	t_{CF}	Input amplitude = 100 mV, at falling edge[Note 2]		1.0		μs
Operating current[Note 3]	I_{CPDD}	During operation			250	μA
	A_{IDDS}	In STOP mode[Note 4]		2.0	20	nA
Resolution of D/A converter for reference voltage generator	R_{ES}			8		bit
Overall error of D/A converter for reference voltage generator	A_{INL}	R_{LOAD} ≥ 4 MΩ			±1.2	%FSR
Operation current of D/A converter for reference voltage generator[Note 3]	I_{DADD}	During operation			5	mA
	A_{IDDS2}	In STOP mode[Note 4]			10	μA

Notes 1. Characteristics of pulse response when ANIm input changes from the comparator reference voltage − 100 mV to the comparator reference voltage + 100 mV

2. Characteristics of pulse response when ANIm input changes from the comparator reference voltage + 100 mV to the comparator reference voltage − 100 mV

3. Six comparators are provided in total. The value shows the operating current per comparator.

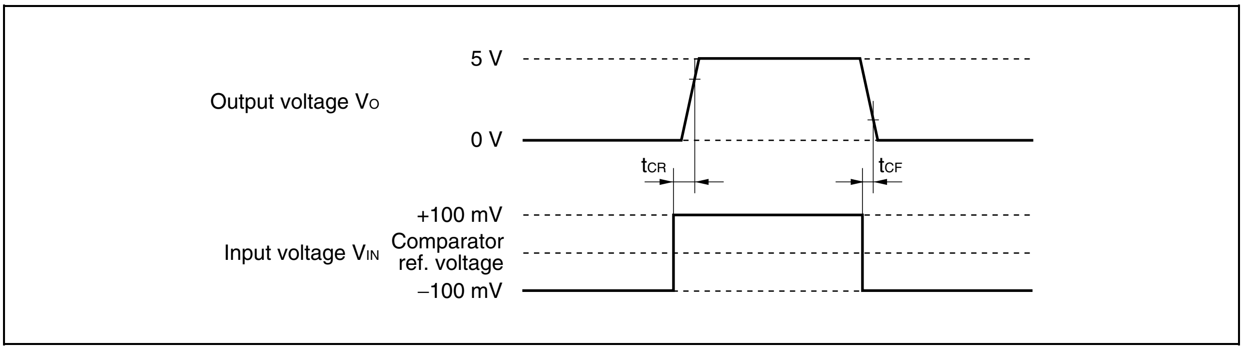
4. Stop comparator operation (CMPnCTL0 register = 00H) before setting STOP mode.

Remarks 1. Power supplies for the comparators are AV_{DD0} and AV_{DD1}.

2. m = 05 to 07, 15 to 17
n = 0, 1

3. R_{LOAD}: Total value of ladder resistor (see **Figures 12-3** and **12-4**.)

Comparator Characteristics



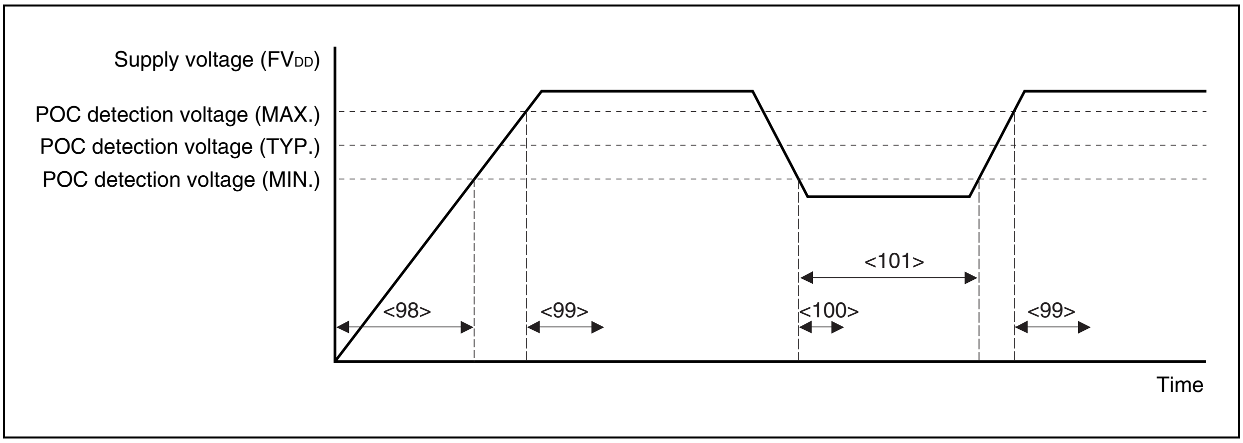
28.2.12 Power-on-clear circuit (POC)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
POC detection voltage	V_{POC0}			3.5	3.7	3.9	V
Supply voltage rise time	t_{PTH}	<98>	FV_{DD} = 0 to 3.5 V	2.5 μs		1.8 s	
Response time 1[Note 1]	t_{PTHD}	<99>	After FV_{DD} reach 3.9 V on power application			3.0	ms
Response time 2[Note 2]	t_{PD}	<100>	After FV_{DD} drop to 3.5 V on power off			1.0	ms
Minimum width of FV_{DD}	t_{PW}	<101>		0.2			ms

Notes 1. The time required to release a reset signal (POCRES) after the POC detection voltage is detected.

2. The time required to output a reset signal (POCRES) after the POC detection voltage is detected.

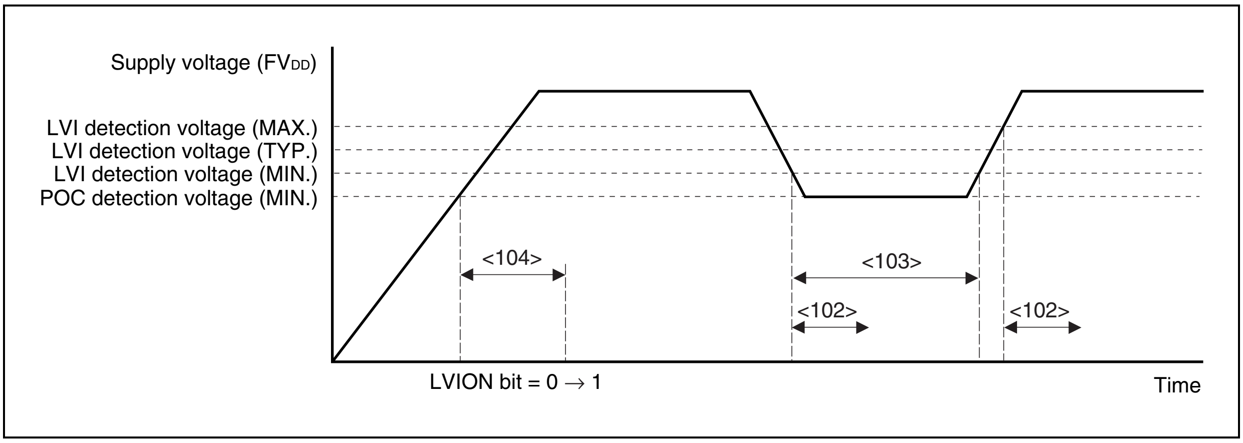

Supply voltage (FVDD)
POC detection voltage (MAX.)
POC detection voltage (TYP.)
POC detection voltage (MIN.)
<98>
<99>
<101>
<100>
<99>
Time

28.2.13 Low-voltage detector (LVI)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = 4.0 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
LVI detection voltage	V_{LVI0}		LVIS.LVIS0 bit = 0	4.2	4.4	4.6	V
	V_{LVI1}		LVIS.LVIS0 bit = 1	4.0	4.2	4.4	V
Response time 1[Note]	t_{LD}	<102>	After FV_{DD} reach V_{LVI0}/V_{LVI1} (MAX.) or drop to V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum width of FV_{DD}	t_{LW}	<103>		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	<104>	After FV_{DD} reach POC detection voltage (MIN.) and the LVIM.LVION bit is changed from 0 to 1		0.1		ms

Note The time required to output an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) after the LVI detection voltage is detected.


Supply voltage (FV_{DD})
LVI detection voltage (MAX.)
LVI detection voltage (TYP.)
LVI detection voltage (MIN.)
POC detection voltage (MIN.)
<104>
<103>
<102>
<102>
LVION bit = 0 → 1
Time

28.2.14 Supply voltage application/cutoff timing

(T_A = −40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD0} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, UV_{DD} = 3.0 to 3.6 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Delay time from FV_{DD} rise to EV_{DD} rise	t_{RER}		−50	0	ms
Delay time from FV_{DD} rise to V_{DD} rise	t_{RVR}		−50	0	ms
Delay time from FV_{DD} rise to AV_{DD} rise	t_{RAR}		−50	0	ms
Delay time from FV_{DD} rise to UV_{DD} rise	t_{RUR}		−50	0	ms
Delay time from FV_{DD} rise to $\overline{RESET}$ rise	t_{RRR}	When using an external reset	T_{osc} + 0.5		ms
Delay time from FV_{DD} fall to EV_{DD} fall	t_{FEF}		0	50	ms
Delay time from FV_{DD} fall to V_{DD} fall	t_{FVF}		0	50	ms
Delay time from FV_{DD} fall to AV_{DD} fall	t_{FAF}		0	50	ms
Delay time from FV_{DD} fall to UV_{DD} fall	t_{FUF}		0	50	ms

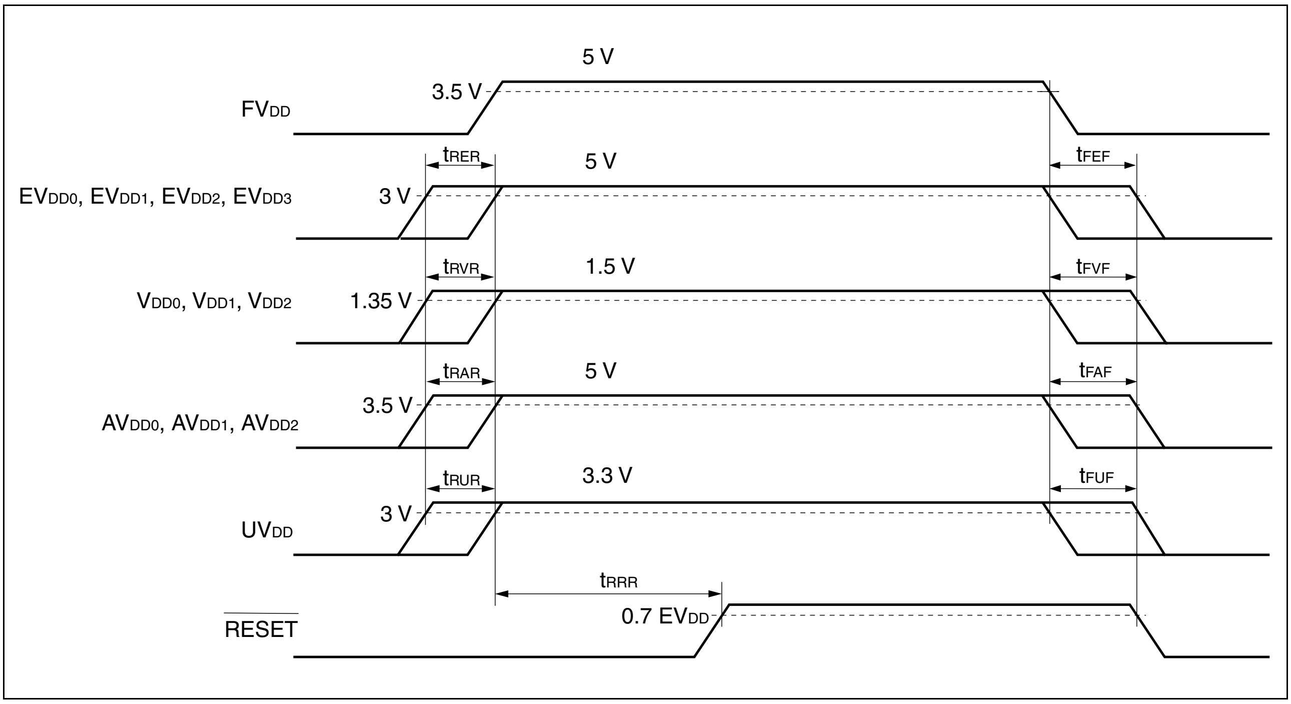
Remark Tosc: Oscillation stabilization time

Supply voltage application/cutoff timing

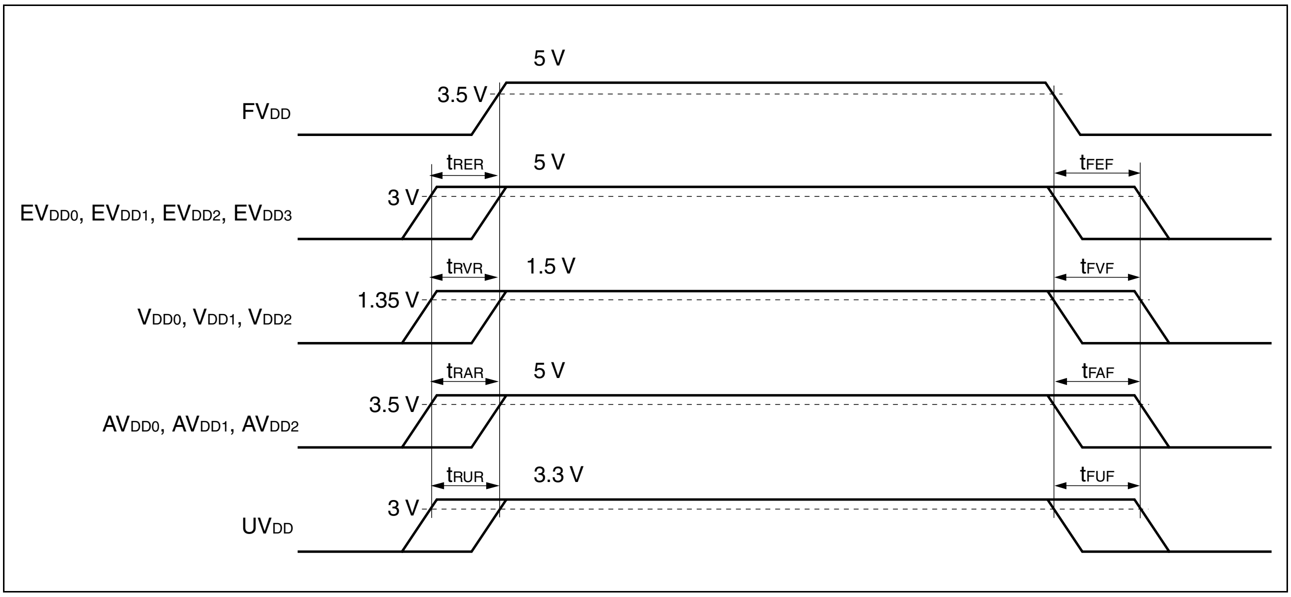
<R> **Cautions 1. There are no regulations for the voltage level and time of FV_{DD}, EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}, V_{DD0}, V_{DD1}, V_{DD2}, AV_{DD0}, AV_{DD1}, AV_{DD2}, and UV_{DD} in the process of natural discharge after power supply cutoff.**

2. Apply all of the FV_{DD}, EV_{DD0}, EV_{DD1}, EV_{DD2}, EV_{DD3}, V_{DD0}, V_{DD1}, V_{DD2}, AV_{DD0}, AV_{DD1}, AV_{DD2}, and UV_{DD} power supplies.
It is prohibited to apply one of these power supplies without supplying them all.

(a) External $\overline{\text{RESET}}$ (recommended conditions)



(b) Internal $\overline{\text{RESET}}$ (recommended conditions)



28.2.15 Flash memory programming characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = EV_{DD3} = 3.0 to 5.5 V, FV_{DD} = 4.0 to 5.5 V, AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = EV_{SS3} = EV_{SS4} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

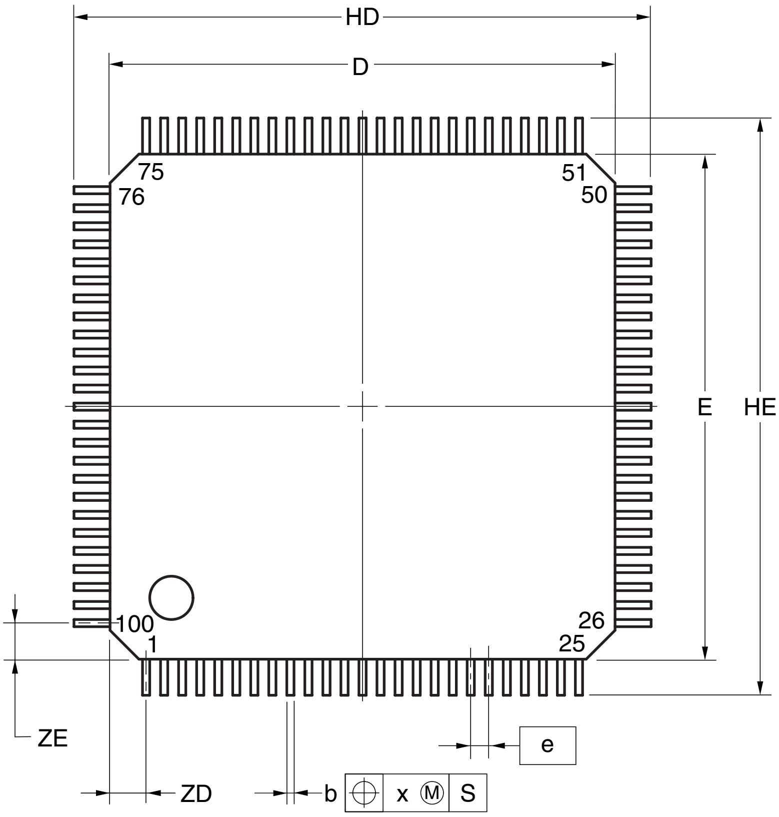
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Rewrite count	C_{ERWR}	**Note**	100			Times

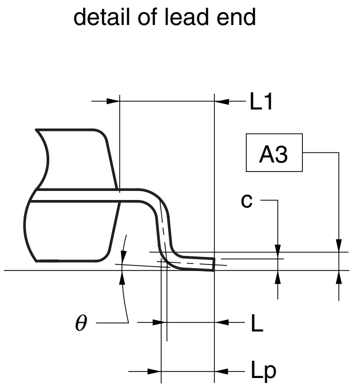
Note Rewrite as follows.

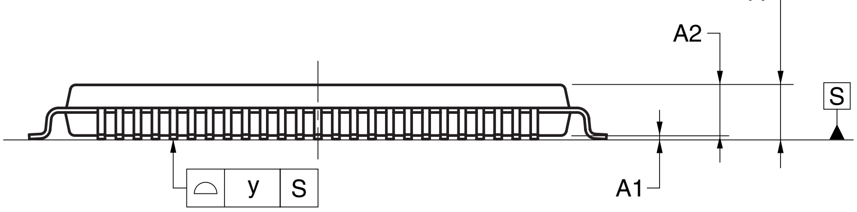
Example when three rewrites: Shipped product →E→P→E→P→E→P (P: Write, E: Erase)

CHAPTER 29 PACKAGE DRAWINGS

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)





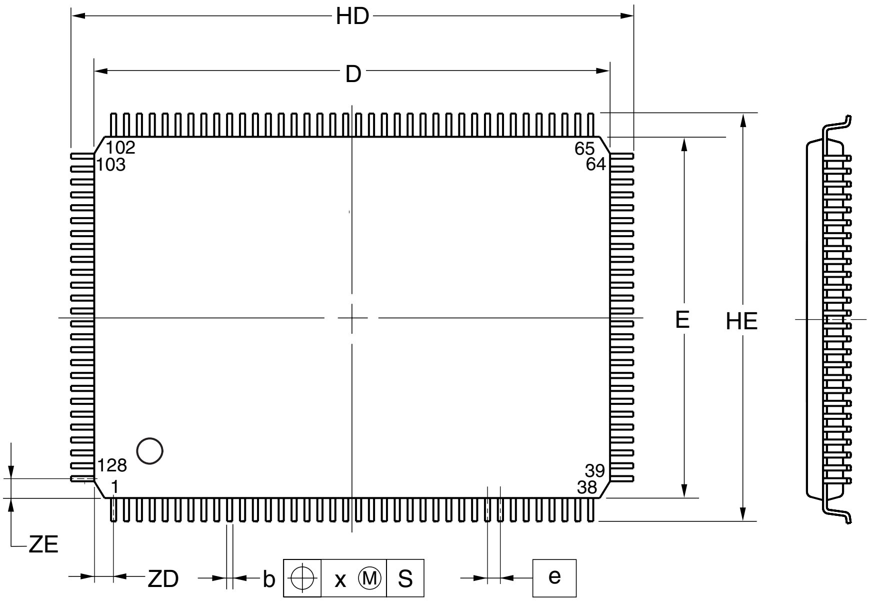


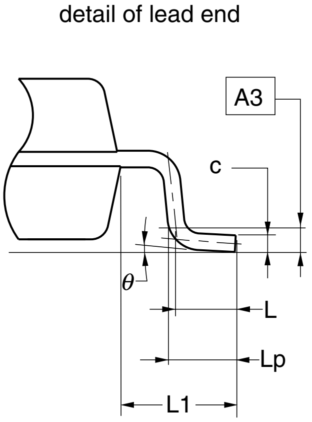
(UNIT:mm)

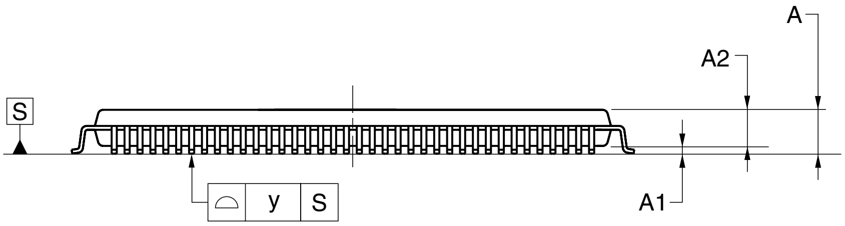
ITEM	DIMENSIONS
D	14.00±0.20
E	14.00±0.20
HD	16.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
$\boxed{A3}$	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
$\boxed{e}$	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

P100GC-50-UEU-1

128-PIN PLASTIC LQFP (FINE PITCH) (14x20)







(UNIT:mm)

ITEM	DIMENSIONS
D	20.00±0.20
E	14.00±0.20
HD	22.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

P128GF-50-GAT

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.
For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.renesas.com/prod/package/manual/index.html)

Table 30-1. Surface Mounting Type Soldering Conditions

μPD70F3919GC-UEU-AX:	100-pin plastic LQFP (fine pitch) (14 × 14 mm)
μPD70F3920GC-UEU-AX:	100-pin plastic LQFP (fine pitch) (14 × 14 mm)
μPD70F3921GC-UEU-AX:	100-pin plastic LQFP (fine pitch) (14 × 14 mm)
μPD70F3922GF-GAT-AX:	128-pin plastic LQFP (fine pitch) (14 × 20 mm)
μPD70F3923GF-GAT-AX:	128-pin plastic LQFP (fine pitch) (14 × 20 mm)
μPD70F3924GF-GAT-AX:	128-pin plastic LQFP (fine pitch) (14 × 20 mm)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR60-107-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	−

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remarks 1. Products with -AX at the end of the part number are lead-free products.

2. For soldering methods and conditions other than those recommended, please contact an Renesas Electronics sales representative.

APPENDIX A CAUTIONS

A.1 Restriction on Conflict Between sld Instruction and Interrupt Request

A.1.1 Description

If a conflict occurs between the decode operation of an instruction in <2> immediately before the sld instruction following an instruction in <1> and an interrupt request before the instruction in <1> is complete, the execution result of the instruction in <1> may not be stored in a register.

Instruction <1>

- ld instruction: ld.b, ld.h, ld.w, ld.bu, ld.hu
- sld instruction: sld.b, sld.h, sld.w, sld.bu, sld.hu
- Multiplication instruction: mul, mulh, mulhi, mulu

Instruction <2>

mov reg1, reg2	not reg1, reg2	satsubr reg1, reg2	satsub reg1, reg2
satadd reg1, reg2	satadd imm5, reg2	or reg1, reg2	xor reg1, reg2
and reg1, reg2	tst reg1, reg2	subr reg1, reg2	sub reg1, reg2
add reg1, reg2	add imm5, reg2	cmp reg1, reg2	cmp imm5, reg2
mulh reg1, reg2	shr imm5, reg2	sar imm5, reg2	shl imm5, reg2

<Example>

```
<i>    ld.w   [r11], r10
              •
              •
              •
<ii>   mov    r10, r28
<iii>  sld.w  0x28, r10
```

If the decode operation of the mov instruction <ii> immediately before the sld instruction <iii> and an interrupt request conflict before execution of the ld instruction <i> is complete, the execution result of instruction <i> may not be stored in a register.

A.1.2 Countermeasure

(1) When compiler (CA850) is used

Use CA850 Ver. 2.61 or later because generation of the corresponding instruction sequence can be automatically suppressed.

(2) For assembler

When executing the sld instruction immediately after instruction <ii>, avoid the above operation using either of the following methods.

- Insert a nop instruction immediately before the sld instruction.
- Do not use the same register as the sld instruction destination register in the above instruction <ii> executed immediately before the sld instruction.

APPENDIX B REGISTER INDEX

(1/24)

Symbol	Name	Unit	Page
AD0CH1	A/D converter 0 channel specification register 1	ADC0	670
AD0CH2	A/D converter 0 channel specification register 2	ADC0	672
AD0CHEN	A/D converter 0 conversion channel specification register	ADC0	662
AD0CHENH	A/D converter 0 conversion channel specification register H	ADC0	662
AD0CHENL	A/D converter 0 conversion channel specification register L	ADC0	662
AD0CR0	A/D0 conversion result register 0	ADC0	664
AD0CR0H	A/D0 conversion result register 0H	ADC0	664
AD0CR1	A/D0 conversion result register 1	ADC0	664
AD0CR1H	A/D0 conversion result register 1H	ADC0	664
AD0CR2	A/D0 conversion result register 2	ADC0	664
AD0CR2H	A/D0 conversion result register 2H	ADC0	664
AD0CR3	A/D0 conversion result register 3	ADC0	664
AD0CR3H	A/D0 conversion result register 3H	ADC0	664
AD0CR4	A/D0 conversion result register 4	ADC0	664
AD0CR4H	A/D0 conversion result register 4H	ADC0	664
AD0CR5	A/D0 conversion result register 5	ADC0	664
AD0CR5H	A/D0 conversion result register 5H	ADC0	664
AD0CR6	A/D0 conversion result register 6	ADC0	664
AD0CR6H	A/D0 conversion result register 6H	ADC0	664
AD0CR7	A/D0 conversion result register 7	ADC0	664
AD0CR7H	A/D0 conversion result register 7H	ADC0	664
AD0CR8	A/D0 conversion result register 8	ADC0	664
AD0CR8H	A/D0 conversion result register 8H	ADC0	664
AD0CR9	A/D0 conversion result register 9	ADC0	664
AD0CR9H	A/D0 conversion result register 9H	ADC0	664
AD0CR10	A/D0 conversion result register 10	ADC0	664
AD0CR10H	A/D0 conversion result register 10H	ADC0	664
AD0CR11	A/D0 conversion result register 11	ADC0	664
AD0CR11H	A/D0 conversion result register 11H	ADC0	664
AD0CR12	A/D0 conversion result register 12	ADC0	664
AD0CR12H	A/D0 conversion result register 12H	ADC0	664
AD0CR13	A/D0 conversion result register 13	ADC0	664
AD0CR13H	A/D0 conversion result register 13H	ADC0	664
AD0CR14	A/D0 conversion result register 14	ADC0	664
AD0CR14H	A/D0 conversion result register 14H	ADC0	664
AD0CR15	A/D0 conversion result register 15	ADC0	664
AD0CR15H	A/D0 conversion result register 15H	ADC0	664
AD0CTC	A/D converter 0 conversion time control register	ADC0	661
AD0CTL0	A/D converter 0 control register	ADC0	668

(2/24)

Symbol	Name	Unit	Page
AD0ECR0	A/D0 conversion result extension register 0	ADC0	674
AD0ECR0H	A/D0 conversion result extension register 0H	ADC0	674
AD0ECR1	A/D0 conversion result extension register 1	ADC0	674
AD0ECR1H	A/D0 conversion result extension register 1H	ADC0	674
AD0ECR2	A/D0 conversion result extension register 2	ADC0	674
AD0ECR2H	A/D0 conversion result extension register 2H	ADC0	674
AD0ECR3	A/D0 conversion result extension register 3	ADC0	674
AD0ECR3H	A/D0 conversion result extension register 3H	ADC0	674
AD0ECR4	A/D0 conversion result extension register 4	ADC0	674
AD0ECR4H	A/D0 conversion result extension register 4H	ADC0	674
AD0FLG	A/D converter 0 flag register	ADC0	676
AD0FLGB	A/D converter 0 flag buffer register	ADC0	677
AD0IC	Interrupt control register	INTC	1213
AD0OCKS	A/D converter 0 clock select register	ADC0	679
AD0SCM	A/D converter 0 scan mode register	ADC0	658
AD0SCMH	A/D converter 0 scan mode register H	ADC0	658
AD0SCML	A/D converter 0 scan mode register L	ADC0	658
AD0TSEL	A/D converter 0 trigger select register	ADC0	669
AD1CH1	A/D converter 1 channel specification register 1	ADC1	670
AD1CH2	A/D converter 1 channel specification register 2	ADC1	672
AD1CHEN	A/D converter 1 conversion channel specification register	ADC1	662
AD1CHENH	A/D converter 1 conversion channel specification register H	ADC1	662
AD1CHENL	A/D converter 1 conversion channel specification register L	ADC1	662
AD1CR0	A/D1 conversion result register 0	ADC1	664
AD1CR0H	A/D1 conversion result register 0H	ADC1	664
AD1CR1	A/D1 conversion result register 1	ADC1	664
AD1CR1H	A/D1 conversion result register 1H	ADC1	664
AD1CR2	A/D1 conversion result register 2	ADC1	664
AD1CR2H	A/D1 conversion result register 2H	ADC1	664
AD1CR3	A/D1 conversion result register 3	ADC1	664
AD1CR3H	A/D1 conversion result register 3H	ADC1	664
AD1CR4	A/D1 conversion result register 4	ADC1	664
AD1CR4H	A/D1 conversion result register 4H	ADC1	664
AD1CR5	A/D1 conversion result register 5	ADC1	664
AD1CR5H	A/D1 conversion result register 5H	ADC1	664
AD1CR6	A/D1 conversion result register 6	ADC1	664
AD1CR6H	A/D1 conversion result register 6H	ADC1	664
AD1CR7	A/D1 conversion result register 7	ADC1	664
AD1CR7H	A/D1 conversion result register 7H	ADC1	664
AD1CR8	A/D1 conversion result register 8	ADC1	664
AD1CR8H	A/D1 conversion result register 8H	ADC1	664
AD1CR9	A/D1 conversion result register 9	ADC1	664
AD1CR9H	A/D1 conversion result register 9H	ADC1	664

(3/24)

Symbol	Name	Unit	Page
AD1CR10	A/D1 conversion result register 10	ADC1	664
AD1CR10H	A/D1 conversion result register 10H	ADC1	664
AD1CR11	A/D1 conversion result register 11	ADC1	664
AD1CR11H	A/D1 conversion result register 11H	ADC1	664
AD1CR12	A/D1 conversion result register 12	ADC1	664
AD1CR12H	A/D1 conversion result register 12H	ADC1	664
AD1CR13	A/D1 conversion result register 13	ADC1	664
AD1CR13H	A/D1 conversion result register 13H	ADC1	664
AD1CR14	A/D1 conversion result register 14	ADC1	664
AD1CR14H	A/D1 conversion result register 14H	ADC1	664
AD1CR15	A/D1 conversion result register 15	ADC1	664
AD1CR15H	A/D1 conversion result register 15H	ADC1	664
AD1CTC	A/D converter 1 conversion time control register	ADC1	661
AD1CTL0	A/D converter 1 control register	ADC1	668
AD1ECR0	A/D1 conversion result extension register 0	ADC1	674
AD1ECR0H	A/D1 conversion result extension register 0H	ADC1	674
AD1ECR1	A/D1 conversion result extension register 1	ADC1	674
AD1ECR1H	A/D1 conversion result extension register 1H	ADC1	674
AD1ECR2	A/D1 conversion result extension register 2	ADC1	674
AD1ECR2H	A/D1 conversion result extension register 2H	ADC1	674
AD1ECR3	A/D1 conversion result extension register 3	ADC1	674
AD1ECR3H	A/D1 conversion result extension register 3H	ADC1	674
AD1ECR4	A/D1 conversion result extension register 4	ADC1	674
AD1ECR4H	A/D1 conversion result extension register 4H	ADC1	674
AD1FLG	A/D converter 1 flag register	ADC1	676
AD1FLGB	A/D converter 1 flag buffer register	ADC1	677
AD1IC	Interrupt control register	INTC	1213
AD1OCKS	A/D converter 1 clock select register	ADC1	679
AD1SCM	A/D converter 1 scan mode register	ADC1	658
AD1SCMH	A/D converter 1 scan mode register H	ADC1	658
AD1SCML	A/D converter 1 scan mode register L	ADC1	658
AD1TSEL	A/D converter 1 trigger select register	ADC1	669
AD2CR0	A/D2 conversion result register 0	ADC2	728
AD2CR0H	A/D2 conversion result register 0H	ADC2	728
AD2CR1	A/D2 conversion result register 1	ADC2	728
AD2CR1H	A/D2 conversion result register 1H	ADC2	728
AD2CR2	A/D2 conversion result register 2	ADC2	728
AD2CR2H	A/D2 conversion result register 2H	ADC2	728
AD2CR3	A/D2 conversion result register 3	ADC2	728
AD2CR3H	A/D2 conversion result register 3H	ADC2	728
AD2CR4	A/D2 conversion result register 4	ADC2	728
AD2CR4H	A/D2 conversion result register 4H	ADC2	728
AD2CR5	A/D2 conversion result register 5	ADC2	728
AD2CR5H	A/D2 conversion result register 5H	ADC2	728

(4/24)

Symbol	Name	Unit	Page
AD2CR6	A/D2 conversion result register 6	ADC2	728
AD2CR6H	A/D2 conversion result register 6H	ADC2	728
AD2CR7	A/D2 conversion result register 7	ADC2	728
AD2CR7H	A/D2 conversion result register 7H	ADC2	728
AD2CR8	A/D2 conversion result register 8	ADC2	728
AD2CR8H	A/D2 conversion result register 8H	ADC2	728
AD2CR9	A/D2 conversion result register 9	ADC2	728
AD2CR9H	A/D2 conversion result register 9H	ADC2	728
AD2CR10	A/D2 conversion result register 10	ADC2	728
AD2CR10H	A/D2 conversion result register 10H	ADC2	728
AD2CR11	A/D2 conversion result register 11	ADC2	728
AD2CR11H	A/D2 conversion result register 11H	ADC2	728
AD2IC	Interrupt control register	INTC	1213
AD2M0	A/D converter 2 mode register 0	ADC2	725
AD2M1	A/D converter 2 mode register 1	ADC2	726
AD2S	A/D converter 2 channel specification register	ADC2	727
ADLTS1	A/D LDTRG1 input select register	ADC0, ADC1	678
ADLTS2	A/D LDTRG2 input select register	ADC0, ADC1	678
ADT0IC	Interrupt control register	INTC	1213
ADT1IC	Interrupt control register	INTC	1213
ADTF	A/D trigger falling edge specification register	ADC0, ADC1	680, 1228
ADTR	A/D trigger rising edge specification register	ADC0, ADC1	680, 1228
AWC	Address wait control register	BCU	1145
BCC	Bus cycle control register	BCU	1148
BCT0	Bus cycle type configuration register 0	BCU	1133
BRGINTE	Bridge interrupt enable register	USBF	1084
BRGINTT	Bridge interrupt control register	USBF	1083
BSC	Bus size configuration register	BCU	1135
CF0CTL0	CSIF0 control register 0	CSIF	848
CF0CTL1	CSIF0 control register 1	CSIF	851
CF0CTL2	CSIF0 control register 2	CSIF	852
CF0REIC	Interrupt control register	INTC	1213
CF0RIC	Interrupt control register	INTC	1213
CF0RX	CSIF0 receive data register	CSIF	846
CF0RXL	CSIF0 receive data register L	CSIF	846
CF0STR	CSIF0 status register	CSIF	854
CF0TIC	Interrupt control register	INTC	1213
CF0TX	CSIF0 transmit data register	CSIF	847
CF0TXL	CSIF0 transmit data register L	CSIF	847
CF1CTL0	CSIF1 control register 0	CSIF	848
CF1CTL1	CSIF1 control register 1	CSIF	851
CF1CTL2	CSIF1 control register 2	CSIF	852
CF1REIC	Interrupt control register	INTC	1213
CF1RIC	Interrupt control register	INTC	1213

(5/24)

Symbol	Name	Unit	Page
CF1RX	CSIF1 receive data register	CSIF	846
CF1RXL	CSIF1 receive data register L	CSIF	846
CF1STR	CSIF1 status register	CSIF	854
CF1TIC	Interrupt control register	INTC	1213
CF1TX	CSIF1 transmit data register	CSIF	847
CF1TXL	CSIF1 transmit data register L	CSIF	847
CF2CTL0	CSIF2 control register 0	CSIF	848
CF2CTL1	CSIF2 control register 1	CSIF	851
CF2CTL2	CSIF2 control register 2	CSIF	852
CF2REIC	Interrupt control register	INTC	1213
CF2RIC	Interrupt control register	INTC	1213
CF2RX	CSIF2 receive data register	CSIF	846
CF2RXL	CSIF2 receive data register L	CSIF	846
CF2STR	CSIF2 status register	CSIF	854
CF2TIC	Interrupt control register	INTC	1213
CF2TX	CSIF2 transmit data register	CSIF	847
CF2TXL	CSIF2 transmit data register L	CSIF	847
CLM	Clock monitor mode register	CG	190
CMP0CTL0	Comparator 0 control register 0	ADC0	682
CMP0CTL1	Comparator 0 control register 1	ADC0	683
CMP0CTL2	Comparator 0 control register 2	ADC0	684
CMP0CTL3	Comparator 0 control register 3	ADC0	685
CMP1CTL0	Comparator 1 control register 0	ADC1	682
CMP1CTL1	Comparator 1 control register 1	ADC1	683
CMP1CTL2	Comparator 1 control register 2	ADC1	684
CMP1CTL3	Comparator 1 control register 3	ADC1	685
CMPIC0F	Interrupt control register	INTC	1213
CMPIC0L	Interrupt control register	INTC	1213
CMPIC1F	Interrupt control register	INTC	1213
CMPIC1L	Interrupt control register	INTC	1213
CMPNFC0F	Comparator output digital noise elimination register 0F	ADC0	686
CMPNFC0L	Comparator output digital noise elimination register 0L	ADC0	686
CMPNFC1F	Comparator output digital noise elimination register 1F	ADC1	686
CMPNFC1L	Comparator output digital noise elimination register 1L	ADC1	686
CMPOF	Comparator output interrupt falling edge specification register	ADC0, ADC1	687
CMPOR	Comparator output interrupt rising edge specification register	ADC0, ADC1	687
CPUBCTL	CPU I/F bus control register	USBF	1086
DA0CS0	D/A converter 0 conversion value setting register 0	ADC0	689
DA0CS1	D/A converter 0 conversion value setting register 1	ADC0	689
DA0M	D/A converter 0 mode register	ADC0	688
DA1CS0	D/A converter 1 conversion value setting register 0	ADC1	689
DA1CS1	D/A converter 1 conversion value setting register 1	ADC1	689
DA1M	D/A converter 1 mode register	ADC1	688
DADC0	DMA addressing control register 0	DMAC	1172
DADC1	DMA addressing control register 1	DMAC	1172

(6/24)

Symbol	Name	Unit	Page
DADC2	DMA addressing control register 2	DMAC	1172
DADC3	DMA addressing control register 3	DMAC	1172
DADC4	DMA addressing control register 4	DMAC	1172
DADC5	DMA addressing control register 5	DMAC	1172
DADC6	DMA addressing control register 6	DMAC	1172
DCHC0	DMA channel control register 0	DMAC	1173
DCHC1	DMA channel control register 1	DMAC	1173
DCHC2	DMA channel control register 2	DMAC	1173
DCHC3	DMA channel control register 3	DMAC	1173
DCHC4	DMA channel control register 4	DMAC	1173
DCHC5	DMA channel control register 5	DMAC	1173
DCHC6	DMA channel control register 6	DMAC	1173
DDAR0	DMA destination address register 0	DMAC	1165
DDAR0H	DMA destination address register 0H	DMAC	1165
DDAR0L	DMA destination address register 0L	DMAC	1165
DDAR1	DMA destination address register 1	DMAC	1165
DDAR1H	DMA destination address register 1H	DMAC	1165
DDAR1L	DMA destination address register 1L	DMAC	1165
DDAR2	DMA destination address register 2	DMAC	1165
DDAR2H	DMA destination address register 2H	DMAC	1165
DDAR2L	DMA destination address register 2L	DMAC	1165
DDAR3	DMA destination address register 3	DMAC	1165
DDAR3H	DMA destination address register 3H	DMAC	1165
DDAR3L	DMA destination address register 3L	DMAC	1165
DDAR4	DMA destination address register 4	DMAC	1165
DDAR4H	DMA destination address register 4H	DMAC	1165
DDAR4L	DMA destination address register 4L	DMAC	1165
DDAR5	DMA destination address register 5	DMAC	1165
DDAR5H	DMA destination address register 5H	DMAC	1165
DDAR5L	DMA destination address register 5L	DMAC	1165
DDAR6	DMA destination address register 6	DMAC	1165
DDAR6H	DMA destination address register 6H	DMAC	1165
DDAR6L	DMA destination address register 6L	DMAC	1165
DEN	DMA enable register	DMAC	1177
DMAIC0	Interrupt control register	INTC	1213
DMAIC1	Interrupt control register	INTC	1213
DMAIC2	Interrupt control register	INTC	1213
DMAIC3	Interrupt control register	INTC	1213
DMAIC4	Interrupt control register	INTC	1213
DMAIC5	Interrupt control register	INTC	1213
DMAIC6	Interrupt control register	INTC	1213
DMAS	DMA status register	DMAC	1176
DMAWC0	DMA wait control register 0	DMAC	91
DMAWC1	DMA wait control register 1	DMAC	91

(7/24)

Symbol	Name	Unit	Page
DMSTP	DMA stop register	DMAC	1178
DSAR0	DMA source address register 0	DMAC	1168
DSAR0H	DMA source address register 0H	DMAC	1168
DSAR0L	DMA source address register 0L	DMAC	1168
DSAR1	DMA source address register 1	DMAC	1168
DSAR1H	DMA source address register 1H	DMAC	1168
DSAR1L	DMA source address register 1L	DMAC	1168
DSAR2	DMA source address register 2	DMAC	1168
DSAR2H	DMA source address register 2H	DMAC	1168
DSAR2L	DMA source address register 2L	DMAC	1168
DSAR3	DMA source address register 3	DMAC	1168
DSAR3H	DMA source address register 3H	DMAC	1168
DSAR3L	DMA source address register 3L	DMAC	1168
DSAR4	DMA source address register 4	DMAC	1168
DSAR4H	DMA source address register 4H	DMAC	1168
DSAR4L	DMA source address register 4L	DMAC	1168
DSAR5	DMA source address register 5	DMAC	1168
DSAR5H	DMA source address register 5H	DMAC	1168
DSAR5L	DMA source address register 5L	DMAC	1168
DSAR6	DMA source address register 6	DMAC	1168
DSAR6H	DMA source address register 6H	DMAC	1168
DSAR6L	DMA source address register 6L	DMAC	1168
DTCR0	DMA transfer times specification register 0	DMAC	1171
DTCR1	DMA transfer times specification register 1	DMAC	1171
DTCR2	DMA transfer times specification register 2	DMAC	1171
DTCR3	DMA transfer times specification register 3	DMAC	1171
DTCR4	DMA transfer times specification register 4	DMAC	1171
DTCR5	DMA transfer times specification register 5	DMAC	1171
DTCR6	DMA transfer times specification register 6	DMAC	1171
DTFR0	DMA trigger factor register 0	DMAC	1179
DTFR0H	DMA trigger factor register 0H	DMAC	1179
DTFR0L	DMA trigger factor register 0L	DMAC	1179
DTFR1	DMA trigger factor register 1	DMAC	1179
DTFR1H	DMA trigger factor register 1H	DMAC	1179
DTFR1L	DMA trigger factor register 1L	DMAC	1179
DTFR2	DMA trigger factor register 2	DMAC	1179
DTFR2H	DMA trigger factor register 2H	DMAC	1179
DTFR2L	DMA trigger factor register 2L	DMAC	1179
DTFR3	DMA trigger factor register 3	DMAC	1179
DTFR3H	DMA trigger factor register 3H	DMAC	1179
DTFR3L	DMA trigger factor register 3L	DMAC	1179
DTFR4	DMA trigger factor register 4	DMAC	1179
DTFR4H	DMA trigger factor register 4H	DMAC	1179
DTFR4L	DMA trigger factor register 4L	DMAC	1179

(8/24)

(9/24)

Symbol	Name	Unit	Page
IMR0	Interrupt mask register 0	INTC	1218
IMR0H	Interrupt mask register 0H	INTC	1218
IMR0L	Interrupt mask register 0L	INTC	1218
IMR1	Interrupt mask register 1	INTC	1218
IMR1H	Interrupt mask register 1H	INTC	1218
IMR1L	Interrupt mask register 1L	INTC	1218
IMR2	Interrupt mask register 2	INTC	1218
IMR2H	Interrupt mask register 2H	INTC	1218
IMR2L	Interrupt mask register 2L	INTC	1218
IMR3	Interrupt mask register 3	INTC	1218
IMR3H	Interrupt mask register 3H	INTC	1218
IMR3L	Interrupt mask register 3L	INTC	1218
IMR4	Interrupt mask register 4	INTC	1218
IMR4H	Interrupt mask register 4H	INTC	1218
IMR4L	Interrupt mask register 4L	INTC	1218
IMR5	Interrupt mask register 5	INTC	1218
IMR5H	Interrupt mask register 5H	INTC	1218
IMR5L	Interrupt mask register 5L	INTC	1218
IMR6	Interrupt mask register 6	INTC	1218
IMR6H	Interrupt mask register 6H	INTC	1218
IMR6L	Interrupt mask register 6L	INTC	1218
INTF0	External interrupt falling edge specification register 0	INTC	1224
INTF1	External interrupt falling edge specification register 1	INTC	1225
INTF2	External interrupt falling edge specification register 2	INTC	1226
INTF3	External interrupt falling edge specification register 3	INTC	1227
INTNFC00	Digital noise elimination 0 control register 00	Port	174
INTNFC01	Digital noise elimination 0 control register 01	Port	174
INTNFC02	Digital noise elimination 0 control register 02	Port	174
INTNFC17	Digital noise elimination 0 control register 17	Port	174
INTNFC18	Digital noise elimination 0 control register 18	Port	174
INTNFC19	Digital noise elimination 0 control register 19	Port	174
INTR0	External interrupt rising edge specification register 0	INTC	1224
INTR1	External interrupt rising edge specification register 1	INTC	1225
INTR2	External interrupt rising edge specification register 2	INTC	1226
INTR3	External interrupt rising edge specification register 3	INTC	1227
ISPR	In-service priority register	INTC	1221
LVIHIC	Interrupt control register	INTC	1213
LVILIC	Interrupt control register	INTC	1213
LVIM	Low-voltage detection register	LVI	1259
LVIS	Low-voltage detection level select register	LVI	1260
OP0CTL0	Operational amplifier 0 control register 0	ADC0	681
OP1CTL0	Operational amplifier 1 control register 0	ADC1	681
OSTS	Oscillation stabilization time select register	CG	189
P0	Port 0 register	Port	101
P1	Port 1 register	Port	107

(10/24)

Symbol	Name	Unit	Page
P2	Port 2 register	Port	113
P3	Port 3 register	Port	119
P4	Port 4 register	Port	125
P5	Port 5 register	Port	130
P7H	Port 7 register H	Port	135
P7L	Port 7 register L	Port	135
P9	Port 9 register	Port	137
PCC	Processor clock control register	Port	186
PDL	Port DL register	Port	141
PDLH	Port DLH register	Port	141
PDLL	Port DLL register	Port	141
PF3	Port 3 function register	Port	123
PFC0	Port 0 function control register	Port	103
PFC1	Port 1 function control register	Port	109
PFC2	Port 2 function control register	Port	115
PFC3	Port 3 function control register	Port	121
PFC4	Port 4 function control register	Port	127
PFC5	Port 5 function control register	Port	131
PFCDL	Port DL function control register	Port	143
PFCDLH	Port DL function control register H	Port	143
PFCDLL	Port DL function control register L	Port	143
PFCE0	Port 0 function control expansion register	Port	103
PFCE1	Port 1 function control expansion register	Port	109
PFCE2	Port 2 function control expansion register	Port	115
PFCE3	Port 3 function control expansion register	Port	121
PFCE4	Port 4 function control expansion register	Port	127
PFCE5	Port 5 function control expansion register	Port	132
PFCEDL	Port DL function control expansion register	Port	144
PFCEDLH	Port DL function control expansion register H	Port	144
PFCEDLL	Port DL function control expansion register L	Port	144
PIC00	Interrupt control register	INTC	1213
PIC01	Interrupt control register	INTC	1213
PIC02	Interrupt control register	INTC	1213
PIC03	Interrupt control register	INTC	1213
PIC04	Interrupt control register	INTC	1213
PIC05	Interrupt control register	INTC	1213
PIC06	Interrupt control register	INTC	1213
PIC07	Interrupt control register	INTC	1213
PIC08	Interrupt control register	INTC	1213
PIC09	Interrupt control register	INTC	1213
PIC10	Interrupt control register	INTC	1213
PIC11	Interrupt control register	INTC	1213
PIC12	Interrupt control register	INTC	1213
PIC13	Interrupt control register	INTC	1213
PIC14	Interrupt control register	INTC	1213

(11/24)

Symbol	Name	Unit	Page
PIC15	Interrupt control register	INTC	1213
PIC16	Interrupt control register	INTC	1213
PIC17	Interrupt control register	INTC	1213
PIC18	Interrupt control register	INTC	1213
PIC19	Interrupt control register	INTC	1213
PLLCTL	PLL control register	CG	185
PM0	Port 0 mode register	Port	101
PM1	Port 1 mode register	Port	107
PM2	Port 2 mode register	Port	113
PM3	Port 3 mode register	Port	119
PM4	Port 4 mode register	Port	125
PM5	Port 5 mode register	Port	130
PM9	Port 9 mode register	Port	137
PMC0	Port 0 mode control register	Port	102
PMC1	Port 1 mode control register	Port	108
PMC2	Port 2 mode control register	Port	114
PMC3	Port 3 mode control register	Port	120
PMC4	Port 4 mode control register	Port	126
PMC5	Port 5 mode control register	Port	131
PMC7H	Port 7 mode control register H	Port	135
PMC7L	Port 7 mode control register L	Port	135
PMC9	Port 9 mode control register	Port	138
PMCDL	Port DL mode control register	Port	143
PMCDLH	Port DL mode control register H	Port	143
PMCDLL	Port DL mode control register L	Port	143
PMDL	Port DL mode register	Port	142
PMDLH	Port DL mode register H	Port	142
PMDLL	Port DL mode register L	Port	142
PRCMD	Command register	CPU	89
PSC	Power save control register	CPU	187, 1042
PSMR	Power save mode register	CPU	188, 1043
PU0	Pull-up resistor option register 0	Port	105
PU1	Pull-up resistor option register 1	Port	111
PU2	Pull-up resistor option register 2	Port	117
PU3	Pull-up resistor option register 3	Port	122
PU4	Pull-up resistor option register 4	Port	128
PU5	Pull-up resistor option register 5	Port	133
PU9	Pull-up resistor option register 9	Port	139
PUDL	Pull-up resistor option register DL	Port	145
PUDLH	Pull-up resistor option register DLH	Port	145
PUDLL	Pull-up resistor option register DLL	Port	145
RESF	Reset source flag register	Reset	1252
SVA0	Slave address register 0	I^2C	909
SYS	System status register	CPU	90

(12/24)

Symbol	Name	Unit	Page
TA0CCIC0	Interrupt control register	INTC	1213
TA0CCIC1	Interrupt control register	INTC	1213
TA0OVIC	Interrupt control register	INTC	1213
TA1CCIC0	Interrupt control register	INTC	1213
TA1CCIC1	Interrupt control register	INTC	1213
TA1OVIC	Interrupt control register	INTC	1213
TA2CCIC0	Interrupt control register	INTC	1213
TA2CCIC1	Interrupt control register	INTC	1213
TA2OVIC	Interrupt control register	INTC	1213
TAA0CCR0	TAA0 capture/compare register 0	TAA	211
TAA0CCR1	TAA0 capture/compare register 1	TAA	213
TAA0CNT	TAA0 counter read buffer register	TAA	215
TAA0CTL0	TAA0 control register 0	TAA	203
TAA0CTL1	TAA0 control register 1	TAA	204
TAA0IOC0	TAA0 I/O control register 0	TAA	206
TAA0OPT0	TAA0 option register 0	TAA	210
TAA1CCR0	TAA1 capture/compare register 0	TAA	211
TAA1CCR1	TAA1 capture/compare register 1	TAA	213
TAA1CNT	TAA1 counter read buffer register	TAA	215
TAA1CTL0	TAA1 control register 0	TAA	203
TAA1CTL1	TAA1 control register 1	TAA	204
TAA1IOC0	TAA1 I/O control register 0	TAA	206
TAA1OPT0	TAA1 option register 0	TAA	210
TAA2CCR0	TAA2 capture/compare register 0	TAA	211
TAA2CCR1	TAA2 capture/compare register 1	TAA	213
TAA2CNT	TAA2 counter read buffer register	TAA	215
TAA2CTL0	TAA2 control register 0	TAA	203
TAA2CTL1	TAA2 control register 1	TAA	204
TAA2IOC0	TAA2 I/O control register 0	TAA	206
TAA2IOC1	TAA2 I/O control register 1	TAA	208
TAA2IOC2	TAA2 I/O control register 2	TAA	209
TAA2OPT0	TAA2 option register 0	TAA	210
TAB0CCR0	TAB0 capture/compare register 0	TAB	318
TAB0CCR1	TAB0 capture/compare register 1	TAB	320
TAB0CCR2	TAB0 capture/compare register 2	TAB	321
TAB0CCR3	TAB0 capture/compare register 3	TAB	323
TAB0CNT	TAB0 counter read buffer register	TAB	324
TAB0CTL0	TAB0 control register 0	TAB	311
TAB0CTL1	TAB0 control register 1	TAB	312
TAB0DTC	TAB0 dead-time compare register	TAB	576
TAB0IOC0	TAB0 I/O control register 0	TAB	313
TAB0IOC1	TAB0 I/O control register 1	TAB	315
TAB0IOC2	TAB0 I/O control register 2	TAB	316
TAB0IOC3	TAB0 I/O control register 3	TAB	582

(13/24)

Symbol	Name	Unit	Page
TAB0OPT0	TAB0 option register 0	TAB	317, 577
TAB0OPT1	TAB0 option register 1	TAB	578
TAB0OPT2	TAB0 option register 2	TAB	579
TAB0OPT3	TAB0 option register 3	TAB	581
TAB1CCR0	TAB1 capture/compare register 0	TAB	318
TAB1CCR1	TAB1 capture/compare register 1	TAB	320
TAB1CCR2	TAB1 capture/compare register 2	TAB	321
TAB1CCR3	TAB1 capture/compare register 3	TAB	323
TAB1CNT	TAB1 counter read buffer register	TAB	324
TAB1CTL0	TAB1 control register 0	TAB	311
TAB1CTL1	TAB1 control register 1	TAB	312
TAB1DTC	TAB1 dead-time compare register	TAB	576
TAB1IOC0	TAB1 I/O control register 0	TAB	313
TAB1IOC1	TAB1 I/O control register 1	TAB	315
TAB1IOC2	TAB1 I/O control register 2	TAB	316
TAB1IOC3	TAB1 I/O control register 3	TAB	582
TAB1OPT0	TAB1 option register 0	TAB	317, 578
TAB1OPT1	TAB1 option register 1	TAB	578
TAB1OPT2	TAB1 option register 2	TAB	579
TAB1OPT3	TAB1 option register 3	TAB	581
TANFC2	Digital noise elimination 1 control register 2	Port	175
TB0CCBIC0	Interrupt control register	INTC	1213
TB0CCIC0	Interrupt control register	INTC	1213
TB0CCIC1	Interrupt control register	INTC	1213
TB0CCIC2	Interrupt control register	INTC	1213
TB0CCIC3	Interrupt control register	INTC	1213
TB0OVBIC	Interrupt control register	INTC	1213
TB0OVIC	Interrupt control register	INTC	1213
TB1CCBIC0	Interrupt control register	INTC	1213
TB1CCIC0	Interrupt control register	INTC	1213
TB1CCIC1	Interrupt control register	INTC	1213
TB1CCIC2	Interrupt control register	INTC	1213
TB1CCIC3	Interrupt control register	INTC	1213
TB1OVBIC	Interrupt control register	INTC	1213
TB1OVIC	Interrupt control register	INTC	1213
TM0CMP0	TMM0 compare register 0	TMM	565
TM0CTL0	TMM0 control register 0	TMM	566
TM0EQIC0	Interrupt control register	INTC	1213
TM1CMP0	TMM1 compare register 0	TMM	565
TM1CTL0	TMM1 control register 0	TMM	566
TM1EQIC0	Interrupt control register	INTC	1213
TM2CMP0	TMM2 compare register 0	TMM	565
TM2CTL0	TMM2 control register 0	TMM	566
TM2EQIC0	Interrupt control register	INTC	1213

(14/24)

Symbol	Name	Unit	Page
TM3CMP0	TMM3 compare register 0	TMM	565
TM3CTL0	TMM3 control register 0	TMM	566
TM3EQIC0	Interrupt control register	INTC	1213
TT0CCIC0	Interrupt control register	INTC	1213
TT0CCIC1	Interrupt control register	INTC	1213
TT0CCR0	TMT0 capture/compare register 0	TMT	443
TT0CCR1	TMT0 capture/compare register 1	TMT	445
TT0CNT	TMT0 counter read buffer register	TMT	447
TT0CTL0	TMT0 control register 0	TMT	428
TT0CTL1	TMT0 control register 1	TMT	429
TT0CTL2	TMT0 control register 2	TMT	431
TT0IECIC	Interrupt control register	INTC	1213
TT0IOC0	TMT0 I/O control register 0	TMT	433
TT0IOC1	TMT0 I/O control register 1	TMT	435
TT0IOC2	TMT0 I/O control register 2	TMT	436
TT0IOC3	TMT0 I/O control register 3	TMT	437
TT0OPT0	TMT0 option register 0	TMT	439
TT0OPT1	TMT0 option register 1	TMT	440
TT0OVIC	Interrupt control register	INTC	1213
TT0TCW	TMT0 counter write register	TMT	447
TT1CCIC0	Interrupt control register	INTC	1213
TT1CCIC1	Interrupt control register	INTC	1213
TT1CCR0	TMT1 capture/compare register 0	TMT	443
TT1CCR1	TMT1 capture/compare register 1	TMT	445
TT1CNT	TMT1 counter read buffer register	TMT	447
TT1CTL0	TMT1 control register 0	TMT	428
TT1CTL1	TMT1 control register 1	TMT	429
TT1CTL2	TMT1 control register 2	TMT	431
TT1IECIC	Interrupt control register	INTC	1213
TT1IOC0	TMT1 I/O control register 0	TMT	433
TT1IOC1	TMT1 I/O control register 1	TMT	435
TT1IOC2	TMT1 I/O control register 2	TMT	436
TT1IOC3	TMT1 I/O control register 3	TMT	437
TT1OPT0	TMT1 option register 0	TMT	439
TT1OPT1	TMT1 option register 1	TMT	440
TT1OVIC	Interrupt control register	INTC	1213
TT1TCW	TMT1 counter write register	TMT	447
TT2CCIC0	Interrupt control register	INTC	1213
TT2CCIC1	Interrupt control register	INTC	1213
TT2CCR0	TMT2 capture/compare register 0	TMT	443
TT2CCR1	TMT2 capture/compare register 1	TMT	445
TT2CNT	TMT2 counter read buffer register	TMT	447
TT2CTL0	TMT2 control register 0	TMT	428
TT2CTL1	TMT2 control register 1	TMT	429

(15/24)

Symbol	Name	Unit	Page
TT2IOC0	TMT2 I/O control register 0	TMT	433
TT2IOC1	TMT2 I/O control register 1	TMT	435
TT2IOC2	TMT2 I/O control register 2	TMT	436
TT2OPT0	TMT2 option register 0	TMT	439
TT2OVIC	Interrupt control register	INTC	1213
TT3CCIC0	Interrupt control register	INTC	1213
TT3CCIC1	Interrupt control register	INTC	1213
TT3CCR0	TMT3 capture/compare register 0	TMT	443
TT3CCR1	TMT3 capture/compare register 1	TMT	445
TT3CNT	TMT3 counter read buffer register	TMT	447
TT3CTL0	TMT3 control register 0	TMT	428
TT3CTL1	TMT3 control register 1	TMT	429
TT3IOC0	TMT3 I/O control register 0	TMT	433
TT3IOC1	TMT3 I/O control register 1	TMT	435
TT3IOC2	TMT3 I/O control register 2	TMT	436
TT3OPT0	TMT3 option register 0	TMT	439
TT3OVIC	Interrupt control register	INTC	1213
TTISL0	TMT0 capture input select register	TMT	442
TTISL1	TMT1 capture input select register	TMT	442
TTNFC0	Digital noise elimination 2 control register 0	Port	174
TTNFC1	Digital noise elimination 2 control register 1	Port	174
TTNFC2	Digital noise elimination 3 control register 2	Port	174
TTNFC3	Digital noise elimination 3 control register 3	Port	174
UA0CTL0	UARTA0 control register 0	UARTA	756
UA0CTL1	UARTA0 control register 1	UARTA	773
UA0CTL2	UARTA0 control register 2	UARTA	774
UA0OPT0	UARTA0 option control register 0	UARTA	758
UA0REIC	Interrupt control register	INTC	1213
UA0RIC	Interrupt control register	INTC	1213
UA0RX	UARTA0 receive data register	UARTA	761
UA0STR	UARTA0 status register	UARTA	759
UA0TIC	Interrupt control register	INTC	1213
UA0TX	UARTA0 transmit data register	UARTA	761
UA1CTL0	UARTA1 control register 0	UARTA	756
UA1CTL1	UARTA1 control register 1	UARTA	773
UA1CTL2	UARTA1 control register 2	UARTA	774
UA1OPT0	UARTA1 option control register 0	UARTA	758
UA1REIC	Interrupt control register	INTC	1213
UA1RIC	Interrupt control register	INTC	1213
UA1RX	UARTA1 receive data register	UARTA	761
UA1STR	UARTA1 status register	UARTA	759
UA1TIC	Interrupt control register	INTC	1213
UA1TX	UARTA1 transmit data register	UARTA	761

(16/24)

Symbol	Name	Unit	Page
UA2CTL0	UARTA2 control register 0	UARTA	756
UA2CTL1	UARTA2 control register 1	UARTA	773
UA2CTL2	UARTA2 control register 2	UARTA	774
UA2OPT0	UARTA2 option control register 0	UARTA	758
UA2REIC	Interrupt control register	INTC	1213
UA2RIC	Interrupt control register	INTC	1213
UA2RX	UARTA2 receive data register	UARTA	761
UA2STR	UARTA2 status register	UARTA	759
UA2TIC	Interrupt control register	INTC	1213
UA2TX	UARTA2 transmit data register	UARTA	761
UBCTL0	UARTB control register 0	UARTB	787
UBCTL2	UARTB control register 2	UARTB	792
UBFIC0	UARTB FIFO control register 0	UARTB	796
UBFIC1	UARTB FIFO control register 1	UARTB	798
UBFIC2	UARTB FIFO control register 2	UARTB	799
UBFIC2H	UARTB FIFO control register 2H	UARTB	799
UBFIC2L	UARTB FIFO control register 2L	UARTB	799
UBFIS0	UARTB FIFO status register 0	UARTB	801
UBFIS1	UARTB FIFO status register 1	UARTB	802
UBRX	UARTB receive data register	UARTB	794
UBRXAP	UARTB receive data register AP	UARTB	794
UBSTR	UARTB status register	UARTB	790
UBTX	UARTB transmit data register	UARTB	793
UCKSEL	USB clock select register	USBF	982
UF0AAS	UF0 active alternative setting register	USBF	1038
UF0ADRS	UF0 address register	USBF	1075
UF0AIFN	UF0 active interface number register	USBF	1037
UF0ASS	UF0 alternative setting status register	USBF	1039
UF0BI1	UF0 bulk in 1 register	USBF	1058
UF0BI2	UF0 bulk in 2 register	USBF	1062
UF0BO1	UF0 bulk out 1 register	USBF	1051
UF0BO1L	UF0 bulk out 1 length register	USBF	1054
UF0BO2	UF0 bulk out 2 register	USBF	1055
UF0BO2L	UF0 bulk out 2 length register	USBF	1058
UF0CIE0	UF0 configuration interface endpoint descriptor register 0	USBF	1081
UF0CIE1	UF0 configuration interface endpoint descriptor register 1	USBF	1081
UF0CIE2	UF0 configuration interface endpoint descriptor register 2	USBF	1081
UF0CIE3	UF0 configuration interface endpoint descriptor register 3	USBF	1081
UF0CIE4	UF0 configuration interface endpoint descriptor register 4	USBF	1081
UF0CIE5	UF0 configuration interface endpoint descriptor register 5	USBF	1081
UF0CIE6	UF0 configuration interface endpoint descriptor register 6	USBF	1081
UF0CIE7	UF0 configuration interface endpoint descriptor register 7	USBF	1081
UF0CIE8	UF0 configuration interface endpoint descriptor register 8	USBF	1081
UF0CIE9	UF0 configuration interface endpoint descriptor register 9	USBF	1081

(17/24)

Symbol	Name	Unit	Page
UF0CIE10	UF0 configuration interface endpoint descriptor register 10	USBF	1081
UF0CIE11	UF0 configuration interface endpoint descriptor register 11	USBF	1081
UF0CIE12	UF0 configuration interface endpoint descriptor register 12	USBF	1081
UF0CIE13	UF0 configuration interface endpoint descriptor register 13	USBF	1081
UF0CIE14	UF0 configuration interface endpoint descriptor register 14	USBF	1081
UF0CIE15	UF0 configuration interface endpoint descriptor register 15	USBF	1081
UF0CIE16	UF0 configuration interface endpoint descriptor register 16	USBF	1081
UF0CIE17	UF0 configuration interface endpoint descriptor register 17	USBF	1081
UF0CIE18	UF0 configuration interface endpoint descriptor register 18	USBF	1081
UF0CIE19	UF0 configuration interface endpoint descriptor register 19	USBF	1081
UF0CIE20	UF0 configuration interface endpoint descriptor register 20	USBF	1081
UF0CIE21	UF0 configuration interface endpoint descriptor register 21	USBF	1081
UF0CIE22	UF0 configuration interface endpoint descriptor register 22	USBF	1081
UF0CIE23	UF0 configuration interface endpoint descriptor register 23	USBF	1081
UF0CIE24	UF0 configuration interface endpoint descriptor register 24	USBF	1081
UF0CIE25	UF0 configuration interface endpoint descriptor register 25	USBF	1081
UF0CIE26	UF0 configuration interface endpoint descriptor register 26	USBF	1081
UF0CIE27	UF0 configuration interface endpoint descriptor register 27	USBF	1081
UF0CIE28	UF0 configuration interface endpoint descriptor register 28	USBF	1081
UF0CIE29	UF0 configuration interface endpoint descriptor register 29	USBF	1081
UF0CIE30	UF0 configuration interface endpoint descriptor register 30	USBF	1081
UF0CIE31	UF0 configuration interface endpoint descriptor register 31	USBF	1081
UF0CIE32	UF0 configuration interface endpoint descriptor register 32	USBF	1081
UF0CIE33	UF0 configuration interface endpoint descriptor register 33	USBF	1081
UF0CIE34	UF0 configuration interface endpoint descriptor register 34	USBF	1081
UF0CIE35	UF0 configuration interface endpoint descriptor register 35	USBF	1081
UF0CIE36	UF0 configuration interface endpoint descriptor register 36	USBF	1081
UF0CIE37	UF0 configuration interface endpoint descriptor register 37	USBF	1081
UF0CIE38	UF0 configuration interface endpoint descriptor register 38	USBF	1081
UF0CIE39	UF0 configuration interface endpoint descriptor register 39	USBF	1081
UF0CIE40	UF0 configuration interface endpoint descriptor register 40	USBF	1081
UF0CIE41	UF0 configuration interface endpoint descriptor register 41	USBF	1081
UF0CIE42	UF0 configuration interface endpoint descriptor register 42	USBF	1081
UF0CIE43	UF0 configuration interface endpoint descriptor register 43	USBF	1081
UF0CIE44	UF0 configuration interface endpoint descriptor register 44	USBF	1081
UF0CIE45	UF0 configuration interface endpoint descriptor register 45	USBF	1081
UF0CIE46	UF0 configuration interface endpoint descriptor register 46	USBF	1081
UF0CIE47	UF0 configuration interface endpoint descriptor register 47	USBF	1081
UF0CIE48	UF0 configuration interface endpoint descriptor register 48	USBF	1081
UF0CIE49	UF0 configuration interface endpoint descriptor register 49	USBF	1081
UF0CIE50	UF0 configuration interface endpoint descriptor register 50	USBF	1081
UF0CIE51	UF0 configuration interface endpoint descriptor register 51	USBF	1081
UF0CIE52	UF0 configuration interface endpoint descriptor register 52	USBF	1081
UF0CIE53	UF0 configuration interface endpoint descriptor register 53	USBF	1081

(18/24)

Symbol	Name	Unit	Page
UF0CIE54	UF0 configuration interface endpoint descriptor register 54	USBF	1081
UF0CIE55	UF0 configuration interface endpoint descriptor register 55	USBF	1081
UF0CIE56	UF0 configuration interface endpoint descriptor register 56	USBF	1081
UF0CIE57	UF0 configuration interface endpoint descriptor register 57	USBF	1081
UF0CIE58	UF0 configuration interface endpoint descriptor register 58	USBF	1081
UF0CIE59	UF0 configuration interface endpoint descriptor register 59	USBF	1081
UF0CIE60	UF0 configuration interface endpoint descriptor register 60	USBF	1081
UF0CIE61	UF0 configuration interface endpoint descriptor register 61	USBF	1081
UF0CIE62	UF0 configuration interface endpoint descriptor register 62	USBF	1081
UF0CIE63	UF0 configuration interface endpoint descriptor register 63	USBF	1081
UF0CIE64	UF0 configuration interface endpoint descriptor register 64	USBF	1081
UF0CIE65	UF0 configuration interface endpoint descriptor register 65	USBF	1081
UF0CIE66	UF0 configuration interface endpoint descriptor register 66	USBF	1081
UF0CIE67	UF0 configuration interface endpoint descriptor register 67	USBF	1081
UF0CIE68	UF0 configuration interface endpoint descriptor register 68	USBF	1081
UF0CIE69	UF0 configuration interface endpoint descriptor register 69	USBF	1081
UF0CIE70	UF0 configuration interface endpoint descriptor register 70	USBF	1081
UF0CIE71	UF0 configuration interface endpoint descriptor register 71	USBF	1081
UF0CIE72	UF0 configuration interface endpoint descriptor register 72	USBF	1081
UF0CIE73	UF0 configuration interface endpoint descriptor register 73	USBF	1081
UF0CIE74	UF0 configuration interface endpoint descriptor register 74	USBF	1081
UF0CIE75	UF0 configuration interface endpoint descriptor register 75	USBF	1081
UF0CIE76	UF0 configuration interface endpoint descriptor register 76	USBF	1081
UF0CIE77	UF0 configuration interface endpoint descriptor register 77	USBF	1081
UF0CIE78	UF0 configuration interface endpoint descriptor register 78	USBF	1081
UF0CIE79	UF0 configuration interface endpoint descriptor register 79	USBF	1081
UF0CIE80	UF0 configuration interface endpoint descriptor register 80	USBF	1081
UF0CIE81	UF0 configuration interface endpoint descriptor register 81	USBF	1081
UF0CIE82	UF0 configuration interface endpoint descriptor register 82	USBF	1081
UF0CIE83	UF0 configuration interface endpoint descriptor register 83	USBF	1081
UF0CIE84	UF0 configuration interface endpoint descriptor register 84	USBF	1081
UF0CIE85	UF0 configuration interface endpoint descriptor register 85	USBF	1081
UF0CIE86	UF0 configuration interface endpoint descriptor register 86	USBF	1081
UF0CIE87	UF0 configuration interface endpoint descriptor register 87	USBF	1081
UF0CIE88	UF0 configuration interface endpoint descriptor register 88	USBF	1081
UF0CIE89	UF0 configuration interface endpoint descriptor register 89	USBF	1081
UF0CIE90	UF0 configuration interface endpoint descriptor register 90	USBF	1081
UF0CIE91	UF0 configuration interface endpoint descriptor register 91	USBF	1081
UF0CIE92	UF0 configuration interface endpoint descriptor register 92	USBF	1081
UF0CIE93	UF0 configuration interface endpoint descriptor register 93	USBF	1081
UF0CIE94	UF0 configuration interface endpoint descriptor register 94	USBF	1081
UF0CIE95	UF0 configuration interface endpoint descriptor register 95	USBF	1081
UF0CIE96	UF0 configuration interface endpoint descriptor register 96	USBF	1081
UF0CIE97	UF0 configuration interface endpoint descriptor register 97	USBF	1081

(19/24)

Symbol	Name	Unit	Page
UF0CIE98	UF0 configuration interface endpoint descriptor register 98	USBF	1081
UF0CIE99	UF0 configuration interface endpoint descriptor register 99	USBF	1081
UF0CIE100	UF0 configuration interface endpoint descriptor register 100	USBF	1081
UF0CIE101	UF0 configuration interface endpoint descriptor register 101	USBF	1081
UF0CIE102	UF0 configuration interface endpoint descriptor register 102	USBF	1081
UF0CIE103	UF0 configuration interface endpoint descriptor register 103	USBF	1081
UF0CIE104	UF0 configuration interface endpoint descriptor register 104	USBF	1081
UF0CIE105	UF0 configuration interface endpoint descriptor register 105	USBF	1081
UF0CIE106	UF0 configuration interface endpoint descriptor register 106	USBF	1081
UF0CIE107	UF0 configuration interface endpoint descriptor register 107	USBF	1081
UF0CIE108	UF0 configuration interface endpoint descriptor register 108	USBF	1081
UF0CIE109	UF0 configuration interface endpoint descriptor register 109	USBF	1081
UF0CIE110	UF0 configuration interface endpoint descriptor register 110	USBF	1081
UF0CIE111	UF0 configuration interface endpoint descriptor register 111	USBF	1081
UF0CIE112	UF0 configuration interface endpoint descriptor register 112	USBF	1081
UF0CIE113	UF0 configuration interface endpoint descriptor register 113	USBF	1081
UF0CIE114	UF0 configuration interface endpoint descriptor register 114	USBF	1081
UF0CIE115	UF0 configuration interface endpoint descriptor register 115	USBF	1081
UF0CIE116	UF0 configuration interface endpoint descriptor register 116	USBF	1081
UF0CIE117	UF0 configuration interface endpoint descriptor register 117	USBF	1081
UF0CIE118	UF0 configuration interface endpoint descriptor register 118	USBF	1081
UF0CIE119	UF0 configuration interface endpoint descriptor register 119	USBF	1081
UF0CIE120	UF0 configuration interface endpoint descriptor register 120	USBF	1081
UF0CIE121	UF0 configuration interface endpoint descriptor register 121	USBF	1081
UF0CIE122	UF0 configuration interface endpoint descriptor register 122	USBF	1081
UF0CIE123	UF0 configuration interface endpoint descriptor register 123	USBF	1081
UF0CIE124	UF0 configuration interface endpoint descriptor register 124	USBF	1081
UF0CIE125	UF0 configuration interface endpoint descriptor register 125	USBF	1081
UF0CIE126	UF0 configuration interface endpoint descriptor register 126	USBF	1081
UF0CIE127	UF0 configuration interface endpoint descriptor register 127	USBF	1081
UF0CIE128	UF0 configuration interface endpoint descriptor register 128	USBF	1081
UF0CIE129	UF0 configuration interface endpoint descriptor register 129	USBF	1081
UF0CIE130	UF0 configuration interface endpoint descriptor register 130	USBF	1081
UF0CIE131	UF0 configuration interface endpoint descriptor register 131	USBF	1081
UF0CIE132	UF0 configuration interface endpoint descriptor register 132	USBF	1081
UF0CIE133	UF0 configuration interface endpoint descriptor register 133	USBF	1081
UF0CIE134	UF0 configuration interface endpoint descriptor register 134	USBF	1081
UF0CIE135	UF0 configuration interface endpoint descriptor register 135	USBF	1081
UF0CIE136	UF0 configuration interface endpoint descriptor register 136	USBF	1081
UF0CIE137	UF0 configuration interface endpoint descriptor register 137	USBF	1081
UF0CIE138	UF0 configuration interface endpoint descriptor register 138	USBF	1081
UF0CIE139	UF0 configuration interface endpoint descriptor register 139	USBF	1081
UF0CIE140	UF0 configuration interface endpoint descriptor register 140	USBF	1081
UF0CIE141	UF0 configuration interface endpoint descriptor register 141	USBF	1081

(20/24)

Symbol	Name	Unit	Page
UF0CIE142	UF0 configuration interface endpoint descriptor register 142	USBF	1081
UF0CIE143	UF0 configuration interface endpoint descriptor register 143	USBF	1081
UF0CIE144	UF0 configuration interface endpoint descriptor register 144	USBF	1081
UF0CIE145	UF0 configuration interface endpoint descriptor register 145	USBF	1081
UF0CIE146	UF0 configuration interface endpoint descriptor register 146	USBF	1081
UF0CIE147	UF0 configuration interface endpoint descriptor register 147	USBF	1081
UF0CIE148	UF0 configuration interface endpoint descriptor register 148	USBF	1081
UF0CIE149	UF0 configuration interface endpoint descriptor register 149	USBF	1081
UF0CIE150	UF0 configuration interface endpoint descriptor register 150	USBF	1081
UF0CIE151	UF0 configuration interface endpoint descriptor register 151	USBF	1081
UF0CIE152	UF0 configuration interface endpoint descriptor register 152	USBF	1081
UF0CIE153	UF0 configuration interface endpoint descriptor register 153	USBF	1081
UF0CIE154	UF0 configuration interface endpoint descriptor register 154	USBF	1081
UF0CIE155	UF0 configuration interface endpoint descriptor register 155	USBF	1081
UF0CIE156	UF0 configuration interface endpoint descriptor register 156	USBF	1081
UF0CIE157	UF0 configuration interface endpoint descriptor register 157	USBF	1081
UF0CIE158	UF0 configuration interface endpoint descriptor register 158	USBF	1081
UF0CIE159	UF0 configuration interface endpoint descriptor register 159	USBF	1081
UF0CIE160	UF0 configuration interface endpoint descriptor register 160	USBF	1081
UF0CIE161	UF0 configuration interface endpoint descriptor register 161	USBF	1081
UF0CIE162	UF0 configuration interface endpoint descriptor register 162	USBF	1081
UF0CIE163	UF0 configuration interface endpoint descriptor register 163	USBF	1081
UF0CIE164	UF0 configuration interface endpoint descriptor register 164	USBF	1081
UF0CIE165	UF0 configuration interface endpoint descriptor register 165	USBF	1081
UF0CIE166	UF0 configuration interface endpoint descriptor register 166	USBF	1081
UF0CIE167	UF0 configuration interface endpoint descriptor register 167	USBF	1081
UF0CIE168	UF0 configuration interface endpoint descriptor register 168	USBF	1081
UF0CIE169	UF0 configuration interface endpoint descriptor register 169	USBF	1081
UF0CIE170	UF0 configuration interface endpoint descriptor register 170	USBF	1081
UF0CIE171	UF0 configuration interface endpoint descriptor register 171	USBF	1081
UF0CIE172	UF0 configuration interface endpoint descriptor register 172	USBF	1081
UF0CIE173	UF0 configuration interface endpoint descriptor register 173	USBF	1081
UF0CIE174	UF0 configuration interface endpoint descriptor register 174	USBF	1081
UF0CIE175	UF0 configuration interface endpoint descriptor register 175	USBF	1081
UF0CIE176	UF0 configuration interface endpoint descriptor register 176	USBF	1081
UF0CIE177	UF0 configuration interface endpoint descriptor register 177	USBF	1081
UF0CIE178	UF0 configuration interface endpoint descriptor register 178	USBF	1081
UF0CIE179	UF0 configuration interface endpoint descriptor register 179	USBF	1081
UF0CIE180	UF0 configuration interface endpoint descriptor register 180	USBF	1081
UF0CIE181	UF0 configuration interface endpoint descriptor register 181	USBF	1081
UF0CIE182	UF0 configuration interface endpoint descriptor register 182	USBF	1081
UF0CIE183	UF0 configuration interface endpoint descriptor register 183	USBF	1081
UF0CIE184	UF0 configuration interface endpoint descriptor register 184	USBF	1081
UF0CIE185	UF0 configuration interface endpoint descriptor register 185	USBF	1081

(21/24)

Symbol	Name	Unit	Page
UF0CIE186	UF0 configuration interface endpoint descriptor register 186	USBF	1081
UF0CIE187	UF0 configuration interface endpoint descriptor register 187	USBF	1081
UF0CIE188	UF0 configuration interface endpoint descriptor register 188	USBF	1081
UF0CIE189	UF0 configuration interface endpoint descriptor register 189	USBF	1081
UF0CIE190	UF0 configuration interface endpoint descriptor register 190	USBF	1081
UF0CIE191	UF0 configuration interface endpoint descriptor register 191	USBF	1081
UF0CIE192	UF0 configuration interface endpoint descriptor register 192	USBF	1081
UF0CIE193	UF0 configuration interface endpoint descriptor register 193	USBF	1081
UF0CIE194	UF0 configuration interface endpoint descriptor register 194	USBF	1081
UF0CIE195	UF0 configuration interface endpoint descriptor register 195	USBF	1081
UF0CIE196	UF0 configuration interface endpoint descriptor register 196	USBF	1081
UF0CIE197	UF0 configuration interface endpoint descriptor register 197	USBF	1081
UF0CIE198	UF0 configuration interface endpoint descriptor register 198	USBF	1081
UF0CIE199	UF0 configuration interface endpoint descriptor register 199	USBF	1081
UF0CIE200	UF0 configuration interface endpoint descriptor register 200	USBF	1081
UF0CIE201	UF0 configuration interface endpoint descriptor register 201	USBF	1081
UF0CIE202	UF0 configuration interface endpoint descriptor register 202	USBF	1081
UF0CIE203	UF0 configuration interface endpoint descriptor register 203	USBF	1081
UF0CIE204	UF0 configuration interface endpoint descriptor register 204	USBF	1081
UF0CIE205	UF0 configuration interface endpoint descriptor register 205	USBF	1081
UF0CIE206	UF0 configuration interface endpoint descriptor register 206	USBF	1081
UF0CIE207	UF0 configuration interface endpoint descriptor register 207	USBF	1081
UF0CIE208	UF0 configuration interface endpoint descriptor register 208	USBF	1081
UF0CIE209	UF0 configuration interface endpoint descriptor register 209	USBF	1081
UF0CIE210	UF0 configuration interface endpoint descriptor register 210	USBF	1081
UF0CIE211	UF0 configuration interface endpoint descriptor register 211	USBF	1081
UF0CIE212	UF0 configuration interface endpoint descriptor register 212	USBF	1081
UF0CIE213	UF0 configuration interface endpoint descriptor register 213	USBF	1081
UF0CIE214	UF0 configuration interface endpoint descriptor register 214	USBF	1081
UF0CIE215	UF0 configuration interface endpoint descriptor register 215	USBF	1081
UF0CIE216	UF0 configuration interface endpoint descriptor register 216	USBF	1081
UF0CIE217	UF0 configuration interface endpoint descriptor register 217	USBF	1081
UF0CIE218	UF0 configuration interface endpoint descriptor register 218	USBF	1081
UF0CIE219	UF0 configuration interface endpoint descriptor register 219	USBF	1081
UF0CIE220	UF0 configuration interface endpoint descriptor register 220	USBF	1081
UF0CIE221	UF0 configuration interface endpoint descriptor register 221	USBF	1081
UF0CIE222	UF0 configuration interface endpoint descriptor register 222	USBF	1081
UF0CIE223	UF0 configuration interface endpoint descriptor register 223	USBF	1081
UF0CIE224	UF0 configuration interface endpoint descriptor register 224	USBF	1081
UF0CIE225	UF0 configuration interface endpoint descriptor register 225	USBF	1081
UF0CIE226	UF0 configuration interface endpoint descriptor register 226	USBF	1081
UF0CIE227	UF0 configuration interface endpoint descriptor register 227	USBF	1081
UF0CIE228	UF0 configuration interface endpoint descriptor register 228	USBF	1081
UF0CIE229	UF0 configuration interface endpoint descriptor register 229	USBF	1081

(22/24)

Symbol	Name	Unit	Page
UF0CIE230	UF0 configuration interface endpoint descriptor register 230	USBF	1081
UF0CIE231	UF0 configuration interface endpoint descriptor register 231	USBF	1081
UF0CIE232	UF0 configuration interface endpoint descriptor register 232	USBF	1081
UF0CIE233	UF0 configuration interface endpoint descriptor register 233	USBF	1081
UF0CIE234	UF0 configuration interface endpoint descriptor register 234	USBF	1081
UF0CIE235	UF0 configuration interface endpoint descriptor register 235	USBF	1081
UF0CIE236	UF0 configuration interface endpoint descriptor register 236	USBF	1081
UF0CIE237	UF0 configuration interface endpoint descriptor register 237	USBF	1081
UF0CIE238	UF0 configuration interface endpoint descriptor register 238	USBF	1081
UF0CIE239	UF0 configuration interface endpoint descriptor register 239	USBF	1081
UF0CIE240	UF0 configuration interface endpoint descriptor register 240	USBF	1081
UF0CIE241	UF0 configuration interface endpoint descriptor register 241	USBF	1081
UF0CIE242	UF0 configuration interface endpoint descriptor register 242	USBF	1081
UF0CIE243	UF0 configuration interface endpoint descriptor register 243	USBF	1081
UF0CIE244	UF0 configuration interface endpoint descriptor register 244	USBF	1081
UF0CIE245	UF0 configuration interface endpoint descriptor register 245	USBF	1081
UF0CIE246	UF0 configuration interface endpoint descriptor register 246	USBF	1081
UF0CIE247	UF0 configuration interface endpoint descriptor register 247	USBF	1081
UF0CIE248	UF0 configuration interface endpoint descriptor register 248	USBF	1081
UF0CIE249	UF0 configuration interface endpoint descriptor register 249	USBF	1081
UF0CIE250	UF0 configuration interface endpoint descriptor register 250	USBF	1081
UF0CIE251	UF0 configuration interface endpoint descriptor register 251	USBF	1081
UF0CIE252	UF0 configuration interface endpoint descriptor register 252	USBF	1081
UF0CIE253	UF0 configuration interface endpoint descriptor register 253	USBF	1081
UF0CIE254	UF0 configuration interface endpoint descriptor register 254	USBF	1081
UF0CIE255	UF0 configuration interface endpoint descriptor register 255	USBF	1081
UFCLR	UF0 CLR request register	USBF	1006
UF0CNF	UF0 configuration register	USBF	1076
UF0DD0	UF0 device descriptor register 0	USBF	1080
UF0DD1	UF0 device descriptor register 1	USBF	1080
UF0DD2	UF0 device descriptor register 2	USBF	1080
UF0DD3	UF0 device descriptor register 3	USBF	1080
UF0DD4	UF0 device descriptor register 4	USBF	1080
UF0DD5	UF0 device descriptor register 5	USBF	1080
UF0DD6	UF0 device descriptor register 6	USBF	1080
UF0DD7	UF0 device descriptor register 7	USBF	1080
UF0DD8	UF0 device descriptor register 8	USBF	1080
UF0DD9	UF0 device descriptor register 9	USBF	1080
UF0DD10	UF0 device descriptor register 1	USBF	1080
UF0DD11	UF0 device descriptor register 11	USBF	1080
UF0DD12	UF0 device descriptor register 12	USBF	1080
UF0DD13	UF0 device descriptor register 13	USBF	1080
UF0DD14	UF0 device descriptor register 14	USBF	1080
UF0DD15	UF0 device descriptor register 15	USBF	1080

(23/24)

Symbol	Name	Unit	Page
UF0DD16	UF0 device descriptor register 16	USBF	1080
UF0DD17	UF0 device descriptor register 17	USBF	1080
UF0DEND	UF0 data end register	USBF	1032
UF0DSCL	UF0 descriptor length register	USBF	1079
UF0DSTL	UF0 device status register L	USBF	1068
UF0E0L	UF0 EP0 length register	USBF	1046
UF0E0N	UF0 EP0NAK register	USBF	998
UF0E0NA	UF0 EP0NAKALL register	USBF	1000
UF0E0R	UF0 EP0 read register	USBF	1045
UF0E0SL	UF0 EP0 status register L	USBF	1069
UF0E0ST	UF0 EP0 setup register	USBF	1047
UF0E0W	UF0 EP0 write register	USBF	1049
UF0E1IM	UF0 endpoint 1 interface mapping register	USBF	1040
UF0E1SL	UF0 EP1 status register L	USBF	1070
UF0E2IM	UF0 endpoint 2 interface mapping register	USBF	1041
UF0E2SL	UF0 EP2 status register L	USBF	1071
UF0E3IM	UF0 endpoint 3 interface mapping register	USBF	1042
UF0E3SL	UF0 EP3 status register L	USBF	1072
UF0E4IM	UF0 endpoint 4 interface mapping register	USBF	1043
UF0E4SL	UF0 EP4 status register L	USBF	1073
UF0E7IM	UF0 endpoint 7 interface mapping register	USBF	1044
UF0E7SL	UF0 EP7 status register L	USBF	1074
UF0EN	UF0 EPNAK register	USBF	1001
UF0ENM	UF0 EPNAK mask register	USBF	1004
UF0EPS0	UF0 EP status 0 register	USBF	1008
UF0EPS1	UF0 EP status 1 register	USBF	1010
UF0EPS2	UF0 EP status 2 register	USBF	1011
UF0FIC0	UF0 FIFO clear 0 register	USBF	1030
UF0FIC1	UF0 FIFO clear 1 register	USBF	1031
UF0GPR	UF0 GPR register	USBF	1034
UF0IC0	UF0 INT clear 0 register	USBF	1025
UF0IC1	UF0 INT clear 1 register	USBF	1026
UF0IC2	UF0 INT clear 2 register	USBF	1027
UF0IC3	UF0 INT clear 3 register	USBF	1028
UF0IC4	UF0 INT clear 4 register	USBF	1029
UF0IF0	UF0 interface 0 register	USBF	1078
UF0IF1	UF0 interface 1 register	USBF	1078
UF0IF2	UF0 interface 2 register	USBF	1078
UF0IF3	UF0 interface 3 register	USBF	1078
UF0IF4	UF0 interface 4 register	USBF	1078
UF0IM0	UF0 INT mask 0 register	USBF	1020
UF0IM1	UF0 INT mask 1 register	USBF	1021
UF0IM2	UF0 INT mask 2 register	USBF	1022
UF0IM3	UF0 INT mask 3 register	USBF	1023
UF0IM4	UF0 INT mask 4 register	USBF	1024

(24/24)

Symbol	Name	Unit	Page
UF0INT1	UF0 interrupt 1 register	USBF	1066
UF0IS0	UF0 INT status 0 register	USBF	1012
UF0IS1	UF0 INT status 1 register	USBF	1014
UF0IS2	UF0 INT status 2 register	USBF	1016
UF0IS3	UF0 INT status 3 register	USBF	1017
UF0IS4	UF0 INT status 4 register	USBF	1019
UF0MODC	UF0 mode control register	USBF	1035
UF0MODS	UF0 mode status register	USBF	1036
UF0SDS	UF0 SNDSIE register	USBF	1005
UF0SET	UF0 SET request register	USBF	1007
UFCTL	USB function control register	USBF	982
UIFIC	Interrupt control register	INTC	1213
UREIC	Interrupt control register	INTC	1213
URIC	Interrupt control register	INTC	1213
UTIC	Interrupt control register	INTC	1213
UTOIC	Interrupt control register	INTC	1213
VSWC	System wait control register	CPU	91
WDTE	Watchdog timer enable register	WDT	643
WDTM	Watchdog timer mode register	WDT	642

APPENDIX C INSTRUCTION SET LIST

C.1 Conventions

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General-purpose registers: Used as source registers.
reg2	General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General-purpose registers: Used mainly to store the remainders of division results and the higher order 32 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the conditions code
sp	Stack pointer (SP)
ep	Element pointer (r30)
listX	X item register list

(2) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list
S	1-bit data that specifies a system register in the register list

(3) Register symbols used in operations

Register Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Halfword	Half word (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Register symbols used in execution clock

Register Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Register symbols used in flag operations

Identifier	Explanation
(Blank)	No change
0	Clear to 0
X	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z/E	0 0 1 0	Z = 1	Zero Equal
NZ/NE	1 0 1 0	Z = 0	Not zero Not equal
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
N	0 1 0 0	S = 1	Negative
P	1 1 0 0	S = 0	Positive
T	0 1 0 1	–	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

C.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	0	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	3 Note 2	3 Note 2	3 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23:16) ll GR[reg2] (31:24) ll GR[reg2] (7:0) ll GR[reg2] (15:8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7:0) ll GR[reg2] (15:8) ll GR [reg2] (23:16) ll GR[reg2] (31:24)		1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))		5	5	5					
CLR1	bit#3, disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		4	4	4	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		4	4	4	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2(return PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	4	4	4					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12,[reg1]	0000011001iiiiiL LLLLLLLLLLLRRRRR **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1]$^{\text{Note 6}}$	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1]$^{\text{Note 6}}$ GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1]$^{\text{Note 6}}$ GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15:0) ll GR[reg2] (31:16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110dddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	3	3	3					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	4	4	4					
JR	disp22	0000011110dddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	3	3	3					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))		1	1	**Note 11**					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)		1	1	**Note 11**					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	**Note 11**					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii IIIIIIIIIIIIIIII	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL[Note 22]	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	2 **Note14**	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	2 **Note14**	2					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2][Note 6]xGR[reg1][Note 6]		1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2][Note 6]xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1][Note 6]ximm16		1	1	2					
MULU[Note 22]	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	2 **Note 14**	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	2 **Note 14**	2					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 **Note 3**	3 **Note 3**	3 **Note 3**				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 **Note 3**	3 **Note 3**	3 **Note 3**				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm Note 15	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) GR[reg in list 12]←Load-memory(sp,Word) sp←sp+4 repeat 2 step above until all regs in list12 is loaded PC←GR[reg1]	n+2 Note 4 Note 17	n+2 Note 4 Note 17	n+2 Note 4 Note 17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	4	4	4	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR[reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory(adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7:0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15:0))	1	1	1					
TRAP	vector	00000111111iiiii 0000000100000000	EIPC ←PC+4(return PC) EIPSW ←PSW ECR.EICC ←Exception code (40H to 4FH, 50H to 5FH) PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH (exception code: 40H to 4FH)) 00000050H (when vector is 10H to 1FH (exception code: 50H to 5FH))	4	4	4					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend(imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend(GR[reg1] (7:0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend(GR[reg1] (15:0))	1	1	1					

Notes 1. dddddddd: Higher 8 bits of disp9.

2. 4 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRR: other than 00000.

6. The lower halfword data only are valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.
rrrrr = regID specification
RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.
IIII: Higher 4 bits of imm9.

14. In the case of reg2 = reg3 (the lower 32 bits of the results are not written in the register) or reg3 = r0 (the higher 32 bits of the results are not written in the register), shortened by 1 clock.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.
01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.
10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.
11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

22. Do not make a combination that satisfies all the following conditions when using the “MUL reg1, reg2, reg3” instruction and “MULU reg1, reg2, reg3” instruction. Operation is not guaranteed when an instruction that satisfies the following conditions is executed.

- Reg1 = reg3
- Reg1 ≠ reg2
- Reg1 ≠ r0
- Reg3 ≠ r0

APPENDIX D REVISION HISTORY

D.1 Major Revisions in This Edition

Page	Description
p. 712	Addition of description to **12.5 Internal Equivalent Circuit**
p. 744	Addition of description to **13.6 Internal Equivalent Circuit**
p. 1352	Addition of **Caution** to **28.1.14 Supply voltage application/cutoff timing**
p. 1388	Addition of **Caution** to **28.2.14 Supply voltage application/cutoff timing**
p. 1430	Addition of **D.2 Revision History of Previous Editions**

<R> D.2 Revision History of Previous Editions

A history of the revisions up to this edition is shown below. “Applied to:” indicates the chapters to which the revision

was applied.

(1/2)

Edition	Description	Applied to:
2nd	• Under development → Mass production	Throughout
	Deletion of description in **1.2 Features**	**CHAPTER 1 INTRODUCTION**
	Modification of description in **Figure 8-51 Basic Timing in Triangular-Wave PWM Output Mode**	**CHAPTER 8 16-BIT TIMER/EVENT COUNTER T (TMT)**
	Modification of description in **Figure 10-6 Timing Chart of 6-Phase PWM Output Mode**	**CHAPTER 10 MOTOR CONTROL FUNCTION**
	Modification of description in **Figure 10-21 Timing of Reflecting Rewritten Value**	
	Modification of description in **10.4.5 (3) When not tuning TAAn**	
	Modification of description in **Figure 10-37 TAAn During Tuning Operation**	
	Modification of description in **10.4.6 (1) Operation under boundary condition (operation when 16-bit counter matches INTTAnCC0 signal)**	
	Deletion of description in **12.1 Features**	**CHAPTER 12 A/D CONVERTERS 0 AND 1**
	Modification of description in **Figure 12-3 Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comparator in A/D Converter 0**	
	Modification of description in **Figure 12-4 Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comparator in A/D Converter 1**	
	Addition of **Figure 12-5 CMPnCTL3 Register Selector Circuit Configuration**	
	Deletion of description in **12.2 (9) AV$_{REFPn}$ pin (n = 0, 1)**	
	Deletion of description in **12.2 (11) AV$_{DDn}$ pin (n = 0, 1)**	
	Deletion of description in **13.1 Features**	**CHAPTER 13 A/D CONVERTER 2**
	Deletion of description in **13.2 (7) AV$_{DD2}$ pin**	
	Modification of description in **18.6.3 (28) UF0 data end register (UF0DEND)**	**CHAPTER 18 USB FUNCTION CONTROLLER (USBF)**
	Modification of description in **19.6.1 (1) Data wait control register 0 (DWC0)**	**CHAPTER 19 BUS CONTROL FUNCTION**
	Modification of description in **19.6.1 (2) Address wait control register (AWC)**	
	Modification of description in **19.7 (1) Bus cycle control register (BCC)**	
	Deletion of description in **26.2.4 Cautions**	**CHAPTER 26 ON-CHIP DEBUG FUNCTION**
	Addition of **27.2 Memory Configuration**	**CHAPTER 27 FLASH MEMORY**
	Addition of description to **27.3 Functional Overview**	
	Modification of description in **27.9 Rewriting by Self Programming**	
	Modification of description in **28.1.3 Operating conditions**	**CHAPTER 28 ELECTRICAL SPECIFICATIONS**
	Modification of description in **28.1.4 Clock oscillator characteristics**	
	Modification of description in **28.1.5 DC characteristics**	
	Modification of description in **28.1.12 Power-on-clear circuit (POC)**	
	Modification of description in **28.1.13 Low-voltage detector (LVI)**	

(2/2)

Edition	Description	
2nd	Addition of **28.1.14 Supply voltage application/cutoff timing**	**CHAPTER 28 ELECTRICAL SPECIFICATIONS**
	Modification of description in **28.1.15 Flash memory programming characteristics**	
	Modification of description in **28.2.3 Operating conditions**	
	Modification of description in **28.2.4 Clock oscillator characteristics**	
	Modification of description in **28.2.5 DC characteristics**	
	Modification of description in **28.2.7 (1) Output signal timing**	
	Modification of description in **28.2.7 (4) Bus timing**	
	Modification of description in **28.2.7 (6) CSIF timing**	
	Modification of description in **28.2.12 Power-on-clear circuit (POC)**	
	Modification of description in **28.2.13 Low-voltage detector (LVI)**	
	Addition of **28.2.14 Supply voltage application/cutoff timing**	
	Modification of description in **28.2.15 Flash memory programming characteristics**	
	Modification of description in **CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS**
	Addition of **APPENDIX D REVISION HISTORY**	**APPENDIX D REVISION HISTORY**

V850E/IG4-H, V850E/IH4-H User’s Manual: Hardware

Publication Date: Rev.3.00 September 30, 2011

Published by: Renesas Electronics Corporation

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1

V850E/IG4-H, V850E/IH4-H

Renesas Electronics Corporation

R01UH0306EJ0300
(Previous Number: U19806EJ2V0UD00)